UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Indoor Localization of Wheeled Robots using Multi-sensor Data Fusion with Event-based Measurements

Payam Nazemzadeh

Advisor

Prof. David Macii

Università degli Studi di Trento

Co-Advisor

Prof. Daniele Fontanelli

Università degli Studi di Trento

December 2016

To everyone who shared their
passion, science and energy with me,
especially my parents

Abstract

In the era in which the robots have started to live and work everywhere and in close contact with humans, they should accurately know their own location at any time to be able to move and perform safely. In particular, large and crowded indoor environments are challenging scenarios for robot accurate and robust localization. The theory and the results presented in this dissertation intend to address the crucial issue of wheeled robots indoor localization by proposing some novel solutions in three complementary ways, i.e. improving robots self-localization through data fusion, adopting collaborative localization (e.g. using the position information from other robots) and finally optimizing the placement of landmarks in the environment once the detection range of the chosen sensors is known.

As far as the first subject is concerned, a robot should be able to localize itself in a given reference frame. This problem is studied in detail to achieve a proper and affordable technique for self-localization, regardless of specific environmental features. The proposed solution relies on the integration of relative and absolute position measurements. The former are based on odometry and on an inertial measurement unit. The absolute position and heading data instead are measured sporadically anytime some landmark spread in the environment is detected. Due to the event-based nature of such measurement data, the robot can work autonomously most of time, even if accuracy degrades. Of course, in order to keep positioning uncertainty bounded, it is important that absolute and relative position data are fused properly. For this reason, four different techniques are analyzed and compared in the dissertation.

As far as the problem of optimal landmark placement is concerned, this is addressed by suggesting a novel and easy-to-use geometrical criterion to maximize the distance between the landmarks deployed over a triangular lattice grid, while ensuring that the absolute position measurement sensors can always detect at least one landmark.

Once the local kinematic state of each robot is estimated, a group of them moving in the same environment and able to detect and communicate with one another can also collaborate to share their position information to refine self-localization results. In the dissertation, it will be shown that this approach can provide some benefits, although performance strongly depends on the metrological features of the adopted sensors as well as on the communication range.

Keywords

[Indoor localization, multi-sensor data fusion, wheeled robots, nonlinear estimation]

Acknowledgements

First of all, I would like to express my gratitude to my advisor, Prof. David Macii, for his patience, guidance and encouragement during the whole PhD program. His fundamental support helped me to grow as a research scientist.

My sincere thanks to Dr. Daniele Fontanelli for his suggestions and precious support. I also would like to thank Prof. Luigi Palopoli for his support during my PhD career.

Special thanks goes to Prof. Peter Händel for the opportunity that he gave me to be a visiting researcher at the Royal Institute of Technology, Stockholm; and to Dr. Isaac Skog and Dr. John-olof Nilsson for the fruitful scientific collaboration while I was there.

Finally, I would like to thank my family, my friends and my colleagues for their endless support.

Payam

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and contribution

Today robots are no longer (and not only) the stiff and dangerous machines used in factory automation throughout the last decades. They have evolved into intelligent and flexible agents that interact closely with their users, improve their quality of life and do not pose significant safety threats. They can be used as entertaining machines, tour guides, cleaning staff and other assistance applications of human activities. A major issue in robotics is safety since robots are often supposed to work in close relationship with humans. Due to the fact that users usually trust and rely on such agents, any kind of mistake or fault can result in some dangers.

Navigation is one of the fundamental challenges in designing mobile robots. Robots navigation requires four key properties [93]:

- *Perception,* i.e. the ability to collect suitable and accurate information about the robot and its surrounding environment,
- *Localization,* i.e. the ability to estimate the position and orientation in a given reference frame,
- *Path planning,* i.e. the ability to plan to reach the wanted destination safely,
- *Guidance,* i.e. the ability to drive the robot over the planned route.

Of course, robot navigation highly depends on how accurately the robot knows its own position at anytime. Since localization is essential to support both path planning and robot guidance, this problem deserves a special attention and will be deeply investigated in this dissertation in the case of robots moving indoors. While the outdoors case can be indeed easily addressed by means of global positioning systems (GPS), this kind of

solution can be hardly used indoors due to various limitations [53]. This has led to a large amount of research in the context of indoor localization in the recent years. The market of indoor location technologies has a very high growth rate and has been predicted to worth $23.13 billion by 2021, at a Compound Annual Growth Rate (CAGR) of 37.4% from 2016 to 2021[1].

Therefore, the contribution of this dissertation is about the planar localization of wheeled robotic agents which are supposed to navigate, work and communicate very close to and in contact with humans in large indoor environments such as airports, stations and shopping malls. In particular, the goal is to find a solution to the localization problem which is sufficiently accurate, trustworthy, reliable and robust and at the same time is affordable and suitable for real environments. For such a purpose, most of the solutions available in the literature have some disadvantages. Some of them are not affordable due to high hardware or computational costs, others are not suitable for real environments because they require a strongly instrumented environment. Several other techniques suffer from robustness and accuracy issues in the presence of dynamic obstacles (e.g. crowd).

For all the reasons mentioned above, this dissertation describes various affordable techniques able to integrate different sensing technologies, as well as estimators of relative and absolute position and heading measurements. In particular, the idea is to rely on relative position tracking techniques running on the mobile agent most of time, while relying occasionally on external landmarks spread in the environment to ensure bounded estimation uncertainty. This approach leads to a challenging position estimation problem due to time varying system observability, that depends on which sensors data are available at every sampling time. In particular, absolute position and orientation measurements are inherently event-based, as they depend on the detection of external landmarks or any other information sent to the agent externally. As a consequence, the placement of landmarks in the environment is prominent and should be also taken into account.

An interesting application area of the proposed approach for indoor localization is Ambient Assisted Living (AAL). As known, the median age in developed countries is expected to grow from 28.6 in 1950 to more than 46.4 in 2050 and the number of elderly people (over 65) will be around three times more than the number of children (less than 15) by 2050 [62]. This will lead to an increasing number of people with some sorts of mobility problems, such as physical impairment, degraded cognitive ability and visual or auditory ability reduction. In this situation, ICT based AAL devices will play a key role to increase elderly people confidence and independence. Various types of AAL devices exist. Some examples are the small wearable devices and service robots such as smart walkers and

[1]MarketsandMarkets report, October 2016, http://www.marketsandmarkets.com/Market-Reports/indoor-positioning-navigation-ipin-market-989.html

smart wheelchairs. The localization specifications and requirements of wheeled robotic AAL devices are definitely compatible with the aim of this research. Therefore, the case study used in this dissertation to experiment the proposed localization techniques is a smart walker.

In such systems, any kind of information about the robot location can be helpful to increase the accuracy and to decrease the cost of the device. In addition, if two or more agents know their location and their relative position, they can exchange and share this information to achieve more accurate localization. Such a *collaborative localization* is another topic covered in this dissertation.

1.2 Structure of the Thesis

The main contributions of this dissertation can be classified into three groups, namely *self-localization*, *landmark placement* and *collaborative localization*.

1.2.1 Self-localization

Each robotic agent moving indoors should be able to localize itself without any centralized information. To this aim, some multi-sensor data fusion techniques explained, parametrized and compared in this dissertation are used to combine odometry, inertial and vision-based measurement data. The vision system relies on sporadic detection of QR codes stuck on the ground and is used to correct the position and orientation estimation. To this aim, after a review of some technologies and estimation techniques used for indoor localization (Chapter 2), the robotic platform used as the case study of this dissertation is described in Chapter 3. Moreover, the proposed sensing technologies, i.e. encoders, an inertial measurement unit (IMU), a passive RFID system and a vision system for QR code detection are described and characterized.

Chapter 4 deals with four possible recursive position estimators for the sensor fusion problem at hand. In particular, an Extended Kalman Filter (EKF), an Unscented Kalman Filter (UKF), a Particle Filter (PF), and an Extended H_∞ Filter (EHF) are defined and their desired parameters are chosen properly. The performance of such estimation techniques is evaluated and compared in Chapter 5. The comparison of various properties shows that the EHF is the best choice for the system at hand and provides the most accurate estimation with acceptable convergence times and computational burden.[2]

[2]Some parts of this work has been published by IEEE, DOI: 10.1109/TMECH.2017.2762598, (c) 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

1.2.2 Landmark Placement

Although the landmark detection events occur sporadically, the occurrence rate highly depends on the distance between the landmarks and the detection range of the corresponding sensor. The larger the distance between landmarks, the less the environment has to be instrumented and the more autonomous the system becomes. However, accuracy generally degrades. This problem is described in Chapter 6 and a solution is proposed assuring that at any location or situation at least one landmark can be detected. The solution is given for sensors with a limited range. In general, the landmark placement problem is an NP-hard problem even if the sensor has an omni-directional detection area [9]. The proposed solution in this dissertation adds another complicated parameter of the angular range,since the solution should take into account not only the position but also the orientation of the agent in a given reference frame.[3]

1.2.3 Collaborative localization

If a number of agents with self-localization ability are moving in the same environment, each agent can inform the others, in a given detection range, about its own location. In this situation, if the relative position of such agents is measurable, it can be used as another source of information to refine localization. This is another kind of event-based position measurements. A distributed solution for such a problem is given is Chapter 7 based on an Interlaced Extended Kalman filter (IEKF).[4]

A summary of all subjects addressed in the dissertation, the main results and an overview of ongoing and future activities are finally reported in Chapter 8.

[3]Some parts of this work has been published by IEEE, DOI: DOI: 10.1109/IPIN.2016.7743631, (c) 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

[4]Some parts of this work has been published by IEEE, DOI: DOI: 10.1109/I2MTC.2016.7520443, (c) 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Chapter 2

Indoor Positioning Technology

A variety of technologies and techniques have been used to localize agents moving indoors which are mainly different due to the positioning system model and available information in the system. Since both system model and available information depend on the measurement technique(s) used in the system, the sensors and the type of information gathered by them have a fundamental role. A large variety of sensors can be used for this purpose observing various kinds of pose or motion in different coordinate frames. According to the sensors variety, a measurement system can observe various variables such as position, orientation, velocity, acceleration or any other information which can be used to identify the agent's location. Moreover, the data gathered by a sensor can be usually explained in one of the three following coordinate frames:

- *Global navigation frame (Earth-centered rotational frame)* is world's reference frame with origin at the center of mass of the earth and global X, Y and Z axes defined through the international reference pole and international reference meridian. GPS measures the location in this frame.

- *Local navigation frame* is a tangent plane defined for a limited-size space. The Z axis is usually selected parallel to the gravity vector while the origin and zero heading are defined by the user. Usually the goal of indoor localization is to find the location of the agent of interest with respect to this frame. Thus, for the systems providing information in other navigational frames, a coordinate transformation is required. Different wireless technologies used for localization, such as ultra-wide band (UWB) or radio frequency identification systems (RFID), magnetic, ultrasonic and initialized odometry systems are among the measurement technologies which can gather information directly in this frame.

- *Local sensor frame* is defined with respect to the sensitivity axis of the sensor. Many types of sensor can measure a relative or absolute position in their own frame. Some

examples of such sensors are optical systems, like cameras, dynamic laser scanners and Inertial measurement units (IMUs).

Note that, in fact, sensor measurement data can be only in its own local frame; however, in some of them, the output we can access from the whole sensing system is given in another frame.

Depending on the way the sensor measurements can result in location estimation, indoor localization technologies can be classified into two groups: *relative localization* and *absolute localization* [39, 14]. Each method has some advantages and disadvantages which should be taken in to account in the localization system design according to the application.

2.1 Relative Localization systems

In relative localization systems, the agent's acceleration, velocity or displacement from one time to another can be observed. In this situation, the position estimation at a given time is related to the position estimations at previous times.

Hence, in relative localization, estimation accuracy depends on the total uncertainty accumulated on the way. The relative localization methods are self-contained, as the agent can estimate its own location autonomously, although a very limited support from external stations is needed to initialize the system. Nonetheless, the incremental process of position estimation leads the estimation error to accumulate unboundedly. Therefore, regardless of the accuracy of the sensor, the pose estimation is not trustworthy after some time.

2.1.1 Odometry

Odometry systems usually measure the distance traveled between any two desired times. Encoders mounted on the wheels are the most common odometry sensors used for wheeled robot localization. They can estimate the displacement by measuring the wheels rotation or the steering orientation [14]. Knowing the system mechanical and geometrical parameters such as wheels radius and the motion model, the displacement and the pose can be simply estimated. Encoders are cheap, fast, reliable and very simple to use. Also, they can give information about position and orientation at the same time. However, the displacement computation depends on the geometrical parameters which may change from agent to agent. Moreover, the position estimation accuracy, highly depends on the wheels mechanical behavior since the encoders cannot detect and measure any undesired mechanical event happening in the wheels like slippage and blockage.

Nowadays, optical sensors are also used for odometry. In visual odometry, the distance traveled over time is estimated by processing sequential camera images [68]. Usually, the features of two or more consecutive images are analyzed and extracted to construct an optical flow field of the motion [88]. Unlike encoders, the visual odometry does not depend on the agent's geometrical parameters. It also can be used in non-wheeled devices and the accuracy of estimation can be much higher than encoders if an appropriate optical flow determination algorithm is available. However, such systems are more complicated and costly than common encoders. Besides, they require a proper fast embedded platform for real-time applications. Such optical systems are used in many applications for localization and tracking, e.g. the NASA's Mars Exploration Rover [20].

The computational cost can be decreased significantly by using two optical sensors and the algorithms which are currently used in optical computer mouse. In practice, the pose can be estimated by the same odometry method as encoders [13]. However, the wheels mechanical behavior affects the estimation less than usual encoders. Unfortunately, like computer mice, they need to continuously track the ground and usually their detection range is very small. Hence, they should be installed very near to the ground which is not feasible in many robotic systems. Moreover, an initial calibration is essential to have good performances.

2.1.2 Inertial Measurement Units

Inertial measurement units (IMUs) are sensing devices which can derive the pose by integrating the measured velocity or acceleration. The development of Micro Electro Mechanical Sensors (MEMS) during the last two decades, has made small and cheap IMUs widely available. This fact as well as their simple installation have made these devices very popular and they are expected to be further used in the future. They are suitable for many robotic systems and can estimate the pose with a simple and lightweight dead reckoning technique. However, the large noise propagation of such MEMS devices results in a quick uncertainty accumulation. Another problem of the MEMS is that their metrological characteristics differ from sensor to sensor. Thus, we can hardly predict estimation uncertainty growth. Besides, an accurate information about the initial position is needed when using such devices. The main sensors used in IMUs are accelerometers, gyroscopes and often magnetometers.

Accelerometers measure the specific force[1] (also called g-force or proper acceleration) of the movement. They often do not directly measure the coordinate acceleration[2]. Moreover, their measurement outputs are given in the sensor local frame. Hence, in order to use

[1]Specific force is the non-gravitational force per unit mass $f_s = \frac{f_{non\text{-}g}}{m}$

[2]Coordinate acceleration is the derivative of velocity

the measured data for localization, a proper method is needed to achieve the coordinate acceleration in the local navigation frame from the specific force data in the local sensor frame. This is a complicated process and many techniques have been introduced to solve this problem using magnetometers or calibrated gyroscopes [22, 47, 107]. After determining the coordinate acceleration in the navigation frame, the position can be identified by integrating the acceleration twice. On the other hand, *gyroscopes* measure the angular rate at every time which can be integrated to estimate the orientation. Again gyroscopes output data is given in the sensors frame and should be referred to the navigation frame to become useful.

2.2 Absolute Localization systems

Absolute localization includes a wide variety of sensing technologies able to directly measure the position and/or orientation with respect to a given frame. Such systems can estimate the agent's pose with a bounded error depending on the sensors' uncertainty. In general, they rely on static or dynamic references, such as beacons, landmarks or satellites with known position and/or orientation in a given reference frame. Hence, the robot localization is not completely autonomous. Moreover, in some situations, the detection of such reference devices may be hard or even impossible.

Global positioning system (GPS), is a common measurement system, usually used outdoors. The GPS can measure the absolute position and time if its receiver mounted on the robot can have unobstructed line of sight connection with at least four global positioning satellites at the same time. Not only the accuracy of GPS data is low for the service robots assuming to move very accurately in public places, but also the GPS signal is too weak to be received indoors. The connectivity problem can be overcome using high sensitivity GPS receivers [89] or Ground-based GPS signal transmitters, usually referred as Pseudolites [103]. a Pseudolite system consists of a location server with GPS reference data and some transmitters to send the location data to the GPS receivers. Although Pseudolites may be able to provide position data continuously, their ability and usage is limited due to some legal and technical issues such as GPS-Pseudolite cross correlation, the receiver saturation and synchronization [48].

Various types of wireless signal techniques such as ultra-wide band (UWB), wifi, ZigBee and radio frequency identification (RFID) have been used for indoor localization. Like GPS, these techniques also need to detect several reference points, such as beacons. After gathering information (usually relative distance and/or direction) from the reference points, a triangulation technique is usually used to estimate the position using lateration or angulation algorithms [58]. In *lateration*, the location is determined by measuring the

agent's distance to various reference points. This distance can be measured using different techniques such as time of arrival (TOA) [42], time difference of arrival (TDOA) [42], received signal strengths (RSS) [66, 75], time of flight (TOF) [37] and phase of arrival [27]. The lateration techniques are significantly affected by the static and dynamic obstacles in the environment and also have several problems due to limited synchronization, multi-path propagate phenomena (especially in crowded environments) and scalability. With *angulation* techniques, the location is estimated as the intersection of the line of sight signals from the agent to various reference points. The angulation techniques do not need the references to be synchronized. Moreover, for 2D positioning, accessibility to two reference points at each time is sufficient. However, the hardware and computational cost of such techniques are high and the estimation error grows significantly if the agent is located far from the reference points [54].

RFID systems have become very popular in the area of localization in the recent years. They are cheap, simple, user-friendly and reliable. RFID tags, usually installed in the walls or on the ground, can be detected rapidly by a reader mounted within the mobile agent [112]. The active RFID tags have large reading range. However, they are more expensive than the passive tags and since a power source is needed inside each tag, the battery should be replaced after a while. Passive RFID tags do not need internal power source, replacement and maintenance and can work for several years. They can provide higher accuracy than the active tags due to their shorter detection range. This characteristic requires to deploy a large number of tags in the environment. Like other wireless techniques, the RFID tags also need direct line of sight connection to the reader, mounted on the agent, to assure optimal performance. This problem can be partially overcome by sticking the passive tags on the ground [74]. In this situation, anytime the agent passes over a tag, there is no obstacle between them. However, an RFID reader cannot provide any information about the orientation. For this reason, Nazari et al. [92] mounted four RFID readers on the agent providing an accurate algorithm to estimate both position and orientation. The solutions based on only passive RFID tags, require sticking a very large number of RFID tags on the ground which may not be practical in large environments.

Ultrasonic sensors can provide very accurate measurements of the distance to reference points [1, 46]. However, similar to other wireless techniques, their performance is affected by the change of crowd and dynamic obstacles.

Positioning systems based on magnetic fields are not affected by multi-path and can operate in non-line of site conditions. Moreover, the use of magnetic fields based technologies typically result in a lower complexity with respect to RF and microwave technologies[77, 76]. However, magnetic positioning systems are characterized by rela-

tively short operating ranges and high power consumption requirements, although the power consumption can be significantly reduced by using resonant systems. Other drawbacks of such systems are the distortion of magnetic fields due to metal objects, the influence of terrain, and the interferences in indoor environments [78].

A large variety of methods for localization is based on optical measurement technologies. The solutions based on image feature extraction typically exploit sequences of images to estimate motion and velocity. Therefore, they should be discussed under relative localization systems. The vision-based absolute positioning systems using static cameras installed in different parts of the environment [109] can be very accurate but have significantly large computational burden and cannot provide decentralized solutions.

The vision-based egomotion techniques are commonly used for Simultaneous Localization and Mapping (SLAM) [36]. Such methods require to detect some natural or artificial landmarks spread in the environment. Natural landmarks commonly exist in the environment, e.g. corners and doors [90]. Artificial landmarks are intentionally deployed in the environment to aid the navigation [35]. The detection of such landmarks is usually easy. However, since the environment should be instrumented previously, this approach cannot be used in the case of unprepared environments. The artificial landmarks can be just arrows showing the direction, special unique shapes or some codes encoding sufficient information for the localization system. A code can be ad-hoc for the application [29, 101] or standard, such as the Quick Response (QR) code [7, 110]. The egomotion solutions are generally quite heavy from the computational point of view (specially the ones based on natural landmarks), suffer from robustness problems due to their sensitivity to changeable light conditions and to the type of landmarks and are sensitive to the landmark initialization errors. Nonetheless, If the landmarks are easy to detect and if their density in the environment is reasonably low, the data fusion of vision-based measurements and dead reckoning could be a viable solution to achieve accurate, scalable and trustworthy localization. The visible light provided by fast flashing LED patterns also is a suitable resource of position information. Such technology provides very accurate localization. However, the environment should be completely instrumented by LEDs which makes a costly and complicated infrastructure implementation.

The absolute localization technologies have a common problem. Since the sensors need to detect some special external reference points, not only the environment should be instrumented properly but also the performance can be severely affected in the presence of different types of static and dynamic obstacles. Hence, many of them, even if provide excellent accuracy in lab experiments, are not appropriate to be used solely for localization in real large and crowded environments. In addition, if the room size is very large, accessibility to walls and doors is very limited.

2.3 Multi-sensor Data Fusion

Each of the relative and absolute position measurement technologies have specific advantages and disadvantages explained shortly in the previous sections. Therefore, since in many applications, using just one technology is not enough, we need to integrate different technologies and use the benefits of each of them. In particular, integrating relative and absolute localization methods can make the system robust, reliable and more accurate as it can benefit from the autonomous property of the relative measurements while keeping position estimation uncertainty bounded. Hence, an appropriate approach for high performance and scalable indoor localization should rely on multi-sensor data fusion [60]. Multi-sensor data fusion is the process of combining data received from several sources to extract maximum possible information from them [50].
In indoor positioning applications based on multi-sensor data fusion two major issues usually arise. First, the most suitable sensing technologies for a given application should be selected and second, an appropriate estimation technique should be used to optimally combine the sensors data.

2.3.1 Sensing technologies for multi-sensor data fusion

All the technologies mentioned as either relative or absolute position measurement sensors in the previous sections can be combined together. A vast amount of research has been done integrating two or more of the available measurement technologies. Each technique has some advantages and drawbacks which should be taken into account in order to select the most suitable one depending on the application. A brief overview of some of the multisensory systems used in the literature is given in Table 2.1.

According to Table 2.1, we can now summarize the main properties and specifications for robotic applications.

Autonomy

The absolute position measurement systems provide bounded localization uncertainty, but require external devices and reference points. If the system autonomy is prominent, an appropriate relative localization system is the best option.

Accuracy

Accuracy is usually a key requirement in every positioning system. None of the relative localization systems can guarantee bounded position estimation uncertainty accumulation and hence they are not reliable for long distance travels. At all, if an accurate system is

Table 2.1: Main features of various position tracking techniques used in the literature

Paper	**Highlights**	**Pros**	**Cons**
[25]	- Odometry and Gyroscope - Sensor data fusion depends on their uncertainties in different conditions of motion	- Very accurate for short run after calibration	- Initialization required - Bounded positioning uncertainty is not guaranteed in the long run
[17]	- Fusion of active sensing and passive RFID reads - Cameras on the ceiling to detect the position of the robot	- Absolute localization - Accurate	- Large number of RFIDs - It relies on external cameras - Difficult to use in large rooms
[16]	Combines global RFID-based positioning with local ultrasound-based distance measurements	- Absolute localization - Very accurate	- Large number of RFID tags - Position refinement difficult in spaces with moving obstacles.
[55]	- Use of encoders - Position correction through RFID mats	- Absolute localization - Mats are easy to deploy and they can be sparse (due to the use of encoders)	- Works well in corridors, but not suitable in large rooms - Unclear data fusion technique - Compass not always suitable in indoor environments
[12]	- Camera-IMU fusion - Vision system uses an egomotion method for optical flow	- Accurate for short and medium distance runs - No artificial landmark to be deployed	- Computationally costly - Unbounded positioning error
[69]	- Camera- Dead reckoning fusion	- Accurate - Obstacle avoidance	- Affected by environment and crowd change - High computational burden

Paper	Highlights	Pros	Cons
[71]	- Camera-IMU fusion - A low computationally cost feature extraction algorithm - On-line IMU calibration	- Accurate for short and medium distance runs - Less computational cost than other optical flow methods - On-line IMU calibration	- Unbounded positioning error - Still high computational cost for a simple embedded platform
[86]	- Camera- Encoder- Gyroscope fusion - Camera reads position information coded inside ceiling-mounted artificial landmarks	- Absolute localization available - less affected by crowd change	- Not standard landmark (using ad hoc custom landmarks) - affected by environment change (height of the ceiling)
[79]	- Magnetic-IMU fusion	- Very low affected by crowd	- Affected by environment change (due to magnetic field change)

needed, with the current available technologies, the usage of absolute position measurement systems is unavoidable. However, to provide some autonomy as well, the fusion of absolute and relative localization techniques seems to be the most appropriate way to localize mobile agents in indoor environments. The selection of the absolute sensing technology highly depends on the required accuracy and the type of environment. In laboratory experiments, the localization uncertainty can range from several meters, to less than few centimeters depending on the chosen technology. However, in real applications, the environmental variations can significantly affect the accuracy.

Environmental features

As shown in Table 2.1, the accuracy of absolute localization technologies that need to detect some features or landmarks by scanning their surrounding horizontal plane, is less in crowded environments simply because obstacles can affect the line of sight connection between the sensor and a landmark or reference point. For instance, the ultrasonic methods can work very accurately in empty places but may not able to give correct measurements in the presence of a large number of obstacles. Therefore, in applications such as ambient assisted living where the system performance should be stable at any time and in every

condition, the most reliable absolute localization systems are the ones based on floor or ceiling landmarks.

The ceiling landmarks are less affected by the presence of crowd than the floor ones. However, they are more affected by the environment changes due to the differences in the ceiling height. Thus, if the same performance is required, the landmarks size should be different according to the ceiling height. Despite this, there exists some environments (e.g. airports and stations) where the ceiling height is larger than the reading range of the adopted sensors. Obviously, another noticeable drawback of the ceiling landmarks is the difficulty of landmarks installation and repair. Therefore, selecting between floor- or ceiling-mounted landmarks requires a kind of trade-off between robustness and simplicity.

2.3.2 Estimation algorithms for Data Fusion

The main reason of using various sensors is to achieve more accurate and reliable results. Hence, a major challenge when dealing with multisensor systems is how to combine the sensor measurements to optimally estimate the desired unknown states. The majority of the methods used for optimal estimation in sensor fusion can be classified into the following two groups [60]

1. *Batch processing* in which all available measurements are processed at the same time.

2. *Sequential (or recursive) processing* in which each measurement is processed and update the state estimation as soon as it becomes available. This is the case of Bayesian estimators.

In batch processing, all the measurements data should be stored. Thus, anytime a new measurement data is received, the estimation problem dimension increases. Such estimators are not very suitable for real-time on-line robot position tracking.
Among the recursive methods, the ones based on *recursive filtering* are the most popular. Depending on the application, measurement methods and sensor noise, different filtering techniques can be used for sensor data fusion.

Kalman filters (KF) are probably the most popular dynamic estimation techniques used for sensor fusion. A KF is a formalized recursive Bayesian technique and is optimal for linear models that rely on inaccurate and uncertain observations with normally distributed zero-mean white noise. A KF produces an estimation of the desired unknown variables minimizing the mean square estimation uncertainty. In other words, Kalman filtering is the process of estimating a-posteriori state vector given the set of measurements up to the current time, with the minimum a-posteriori estimation error covariance. However, to be the optimal estimator, the following assumptions should hold: [6]

- The system is linear;
- The mean and covariance of the initial state, $\hat{s}_{0|0}$ and $P_{0|0}$, are reasonably known;
- The process and measurement noises are Gaussian and white with zero-mean and known covariances;
- The initial state has no correlation with the process and measurement noises;

A variety of other kinds of filters have been designed to address the case when any one of the assumptions above, especially linearity, does not hold. The extended Kalman filters (EKF) are well-known nonlinear Bayesian filters. They can be used either to merge heterogeneous odometer and inertial measurement data [94], or to mitigate the uncertainty growth due to Inertial Measurement Units (IMU) through additional contextual information [18].

In the case of strongly nonlinear systems, the unscented Kalman filters (UKFs) generally provide better results than EKFs, as they usually rely directly on the nonlinear model. UKFs can be effectively applied to problems where posterior distributions are unimodal and they can be fitted, at each time step, by a Gaussian model using a limited number of state space points. Unfortunately, when the process and/or the measurement noise are far from being normally distributed or white, the performance of such dynamic estimators can be far from optimal and estimation accuracy becomes hardly predictable. In such cases, an effective alternative is the Particle filter (PF). A PF estimates the posterior distributions of the internal state variables of a dynamic system heuristically by using a genetic mutation-selection sampling approach. In this way, the relevant points of the state space for given noisy or partial observations are selected adaptively, without any a-priori knowledge about the state-space model or noise distributions [41]. Unfortunately, the inherent heuristic nature of this class of algorithms does not allow to keep the maximum estimation error under control. In addition, PF computational complexity usually grows with the number of particles. Therefore, a trade-off between convergence time and processing burden has to be found. A possible solution to this problem is offered by the H_∞ filters (HF). Similarly to PFs, HFs do not rely on the upfront knowledge of noise distributions and do not require noises to be white, but are purposely designed to minimize the worst-case estimation error with maximum likelihood [95]. The usage of HFs has become increasingly interesting over the last few years [105, 45]. Although the normal HFs provide linear solution, the theory can be extended for nonlinear systems using similar approaches of Kalman filters to achieve extended H_∞ filters (EHFs) [108] or unscented H_∞ filters (UHFs) [57]. In conclusion, the estimation method used in the system highly depends on the available measurements and should be studied separately for

each sensor fusion system. In this dissertation, various kinds of filters (e.g. EKF, UKF, PF and EHF) have been designed, implemented and compared to find their advantages and disadvantages for wheeled robots position tracking.

Chapter 3

Platform Description and Sensor Characterization

In general, the localization problem requires to estimate the position and heading of an agent in a given environment using both a-priori information and heterogeneous sensors data. If the agent of interest is dynamic, like a mobile robot, its location changes over time. As described in previous chapters, the solutions proposed in this dissertation are based on the integration of various relative and absolute position measurement technologies. The robot should be able to estimate its planar position and heading with high accuracy in large indoor environments even in the presence of dynamic obstacles. This is a challenging problem due to the fact that in large crowded environments usual reference points, such as walls, columns and even sometimes ceilings are out of the sensors range. A suitable robotic platform to study this problem is the smart walking assistant, c-Walker, developed in the EU 7th framework programme project DALi[1]. The c-Walker (Figure 3.1) is supposed to support older adults with some mobility or cognitive problems when they move in large public environments, such as airports or shopping malls. As AAL devices are used by people with different types of disabilities, assuring high position accuracy is of crucial importance. Therefore, the c-Walker offers the opportunity to analyze the benefits of combined relative and absolute localization techniques in a key case study. In particular, the system should be designed according to the following specifications:

- The position estimation uncertainty should be smaller than 1 m with 99% probability.
- The localization system should be mainly autonomous , i.e. it should rely on the external absolute measurements as less as possible.

[1]Devices for Assisted Living, www.ict-dali.eu

Figure 3.1: The smart cognitive walking assistant (c-Walker)

- Position tracking accuracy and scalability should be marginally affected by environmental features.
- The computational burden should be low enough to ensure real-time performance using low-cost embedded platforms.
- The cost of the overall system should be as low as possible for the desired accuracy.

This chapter describes the c-Walker prototype, hardware and the measurement technologies used for the localization. The system model, localization algorithms and performance analysis are instead reported in Chapters 4 and 5.

3.1 The c-Walker prototype

A general architectural overview of the c-Walker is shown in Figure 3.2. Users can select a destination in a given map using a touchscreen and a graphic interface. In general, the c-Walker is supposed to provide the user with navigation systems and methods to analyze the environment situation, estimate the current location, plan the optimal route to the destination and suggest the optimal trajectory to reach the destination. The Guidance is based on a haptic or an audio system [70].

The sensors used for localization have been selected according to the specifications listed above. In addition to the sensors for localization purposes, a front Kinect is used for object recognition, anomaly detection and obstacle tracking. Some sensors measurements are transferred to the embedded platform via a CAN bus system while the others (optical sensors) are connected through USB-3 ports.

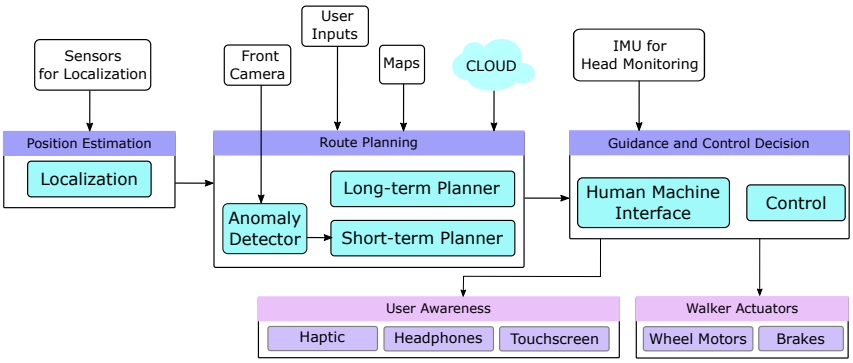


Figure 3.2: Functional block diagram of the c-Walker

The main processing platform for robot's localization, is an *ARM BeagleBone white* embedded platform. Some other algorithms, such as image processing, run on an *Intel Nuc mini-PC*. The specifications of both platforms are shown in table 3.1.

Table 3.1: Main components of the embedded platforms used in the c-Walker

	Item	Features
Sensors and localization	Platform	Beagle Bone White embedded board
	Processor	AM335x ARM Cortex-A8 at 720 MHz
	Memory	256-MB DDR2 RAM
	Storage	8-GB Secure Digital (SD)
	OS	Ubuntu Linux 14.04 LTS
Other algorithms	Platform	Intel Nuc mini-PC
	Processor	Intel I7 5557U at 3.40 GHz
	Memory	8-GB DDR3 RAM
	Storage	256-GB solid state drive (SSD)
	OS	Ubuntu Linux 14.04 LTS

3.2 Sensing technologies for the localization

The basic idea to provide the accurate localization in large indoor environments is to rely mainly on continuous relative localization techniques while adjusting position and orientation with data obtained from floor or ceiling landmarks. We discarded ceiling landmark systems because *a*) the system is supposed to be deployed in large places such as airports where the ceiling height may differ significantly from one room to the other, *b*) the installation is more difficult and complicated. Among various technologies which

can detect some landmarks stuck on the floor, those explored in this research are, *i*) a Radio Frequency Identification (RFID) system detecting and reading passive tags and *ii*) a vision system detecting and reading Quick Response (QR) codes. Such sensors are required to complement the data measured by odometers (e.g. encoders) and an IMU. Of course, in all cases, the metrological characteristics of all sensors should be properly analyzed, regardless of the chosen localization algorithm.

In the following, an overview of the sensors used for the position tracking as well as their characterization is provided.

3.2.1 Incremental encoders

In every wheeled robot, a common technique for position tracking is encoders-based odometry. Incremental encoders provide incremental counts related to the rotation of shaft. The encoders are usually connected to the wheels shaft through a gearing system. Hence, from the counting number, the total rotation of the wheels can be estimated as

$$\Phi_w = \frac{C}{N * G} * 2\pi + \varepsilon^{\Phi} \tag{3.1}$$

where Φ_w is the total distance rotation of the wheel in rad, C is the number of pulses counted by the encoder, N is the number of pulses per revolution (PPR) factor, G is the gear ratio and ε is the uncertainty associated with the encoder measurements. Given a sampling time T_s, the wheel displacement $\Delta\Phi_w$ and angular velocity ω_w at time kT_s ($k \in \mathbb{N}$) are

$$\begin{aligned} \Delta\Phi_{w,k} &= \Phi_{w,k} - \Phi_{w,k-1} \\ \omega_{w,k} &= \Delta\Phi_{w,k}/T_s \end{aligned} \tag{3.2}$$

In the c-Walker, a CUI Inc. AMT10X incremental encoder set to 2048 *ppr* is installed at each rear wheel, with gear ratio of four, to measure the angular velocity of the right and left wheel, i.e. $\omega_{r,k}$ and $\omega_{l,k}$, respectively.

Encoders characterization

To characterize the encoders, the optical incremental encoders were installed on the rear wheels of the Bluebot[2] and several data records were collected with the robot moving along both straight and circular paths. The displacement and time values of the encoders were measured at known positions marked by some stickers put on the floor of our laboratory. Since the robot's velocity in each experiment is constant, knowing the time and distance passed from one reference point to the other, the actual encoders displacement at each

[2]Bluebot is a unicycle robot developed in the Embedded Electronics systems lab, university of Trento, in order to emulate the unicycle-like robotic projects of the lab, such as the smart walker.

sampling time can be estimated and compared with the values returned by the encoders. In all experiments the measurement uncertainty, ε^{Φ_r} and ε^{Φ_l}, between the angular displacements measured by both right and left wheel encoders and the corresponding values estimated at every reference point were calculated. The mean and standard deviation of ε^{Φ_r} and ε^{Φ_l} were computed and plotted as a function of the angular displacements of either wheels. Such plots are shown in Figures 3.3 and 3.4. In the figures, to determine the mean and standard deviation at each displacement value, a sufficient number of points, ranging from around 100 to several thousands have been used.

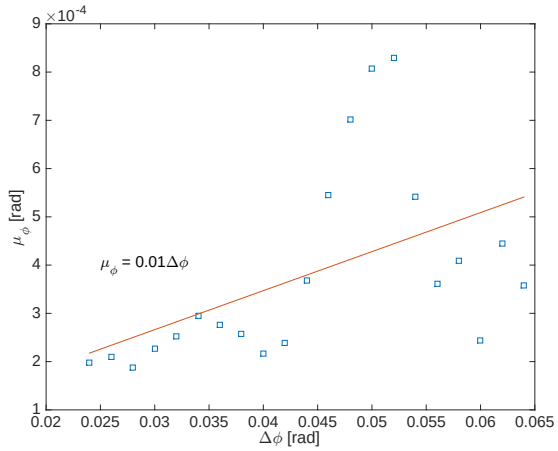


Figure 3.3: Encoders displacements measurement bias as a function of the actual displacement.

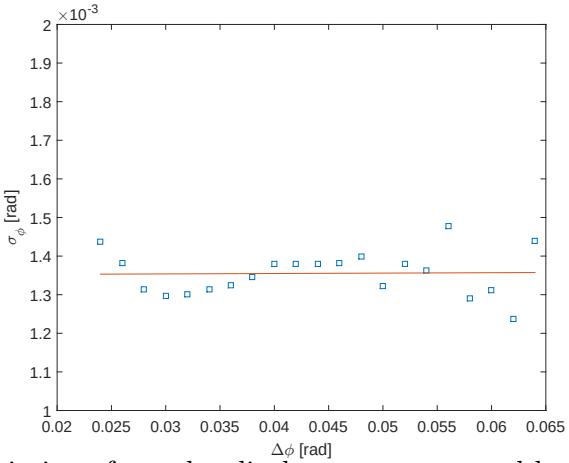


Figure 3.4: Standard deviation of angular displacement measured by encoders as a function of the actual displacement.

The mean of ε^{Φ_r} and ε^{Φ_l} grows with angular displacement. To a fast approximation, such a bias can be estimated with a linear fitting as shown in Figure 3.3, i.e.

$$\begin{cases} \mu_{\Phi_r,k} = 0.01\Delta\Phi_{r,k} \text{ [rad]} \\ \mu_{\Phi_l,k} = 0.01\Delta\Phi_{l,k} \text{ [rad]} \end{cases} \tag{3.3}$$

On the other hand, as shown in Figure 3.4, the standard deviation of angular displacements is quite independent of the actual angular displacements. Therefore, to a fast approximation, we can assume that the standard deviation is constant and can be set to

$$\sigma_{\Phi_r,k} = \sigma_{\Phi_l,k} = 1.35 \cdot 10^{-3} \text{ rad} \tag{3.4}$$

3.2.2 Inertial Measurement Units

In normal situations, odometry based on encoders can be reliable and gives accurate position estimation over short distances. However, in some situations where two wheels do not behave properly or similarly for some time (e.g. when a wheel blocks or slips), the odometry heading estimation error may grow considerably leading the pose estimation to be inaccurate after a while. Thus, to allow the system to work autonomously for a longer time and to prevent the heading estimation to drift away suddenly, it is advisable to measure robot orientation with an IMU as well. Although the accuracy of MEMS IMUs is usually quite low, they still can bring some benefits. In our system, we are just interested in estimating the yaw angle, i.e. the robot's heading, using the IMU.

The gyroscope provides angular velocity data which can be integrated to estimate the orientation. Since the output of the gyroscope is given in its local sensor frame, a method to rotate and to refer such data to the navigation frame is needed [23]. After calibration and rotation, the z-vector of the angular velocity ω^g can be integrated to estimate the heading angle θ^g provided that the initial heading is known. Hence the gyroscope measurement model used in this system is

$$\dot{\theta}^g = \omega^g + \varepsilon^g \tag{3.5}$$

or in discrete form

$$\theta^g_k = \theta^g_{k-1} + T^g_s(\omega^g_k + \varepsilon^g_k) \tag{3.6}$$

where T^g_s is the inertial platform sampling time and ε^g is the uncertainty of the angular velocity estimation using the IMU.

In experiments with the system at hand, an IMU including an Invensense 3000 3-axis gyroscope, a Bosch BMA180 3-axis accelerometer and a Honeywell HMC5883 3-axis magnetometer are used [22]. The inertial measurements data is processed by a 32bit ARM Cortex M3 microelectronic (MCU) running at 72 MHz, with 64 kB of RAM memory and 512 kB of flash memory. As a result, the localization system can just use the orientation estimates returned by the IMU platform.

Gyroscope characterization

The accuracy of the gyroscopic platform was evaluated by means of a calibrated orbital rotator Stuart SB3. The IMU was put on the rotator in 6 different directions to characterize all three axes of the Gyroscope. First the angular velocity values measured by the gyroscopic platform for different fixed angular velocities of the rotator were collected. Then, the histograms showing the velocity uncertainty in different stationary conditions were built. The data analysis showed that in all cases the noise patterns exhibit a white power spectral density and a normal distribution. However, as can be seen in figures 3.5 and 3.6, both the mean value μ_g and the standard deviation σ_g of such distributions tend to grow with the angular velocity ω. The slope of μ_g is quite large and, to a first approximation, it can be roughly regarded as constant. This result means that the systematic error introduced by the gyroscope can be approximated by a linear model as a function of angular velocity (ω) estimated by the gyroscope. Furthermore, even if the noise distribution is globally normal, the value of σ_g is inherently non-stationary due to its dependence on ω.

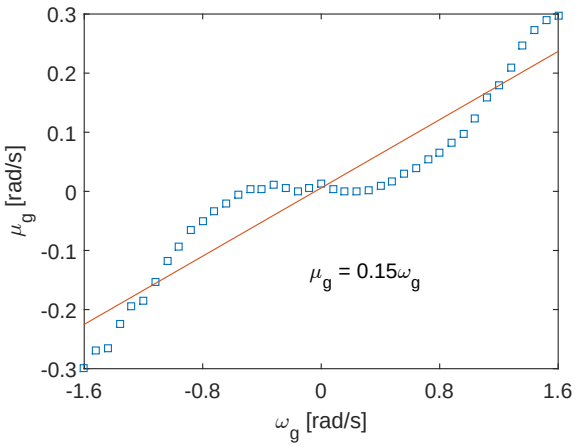


Figure 3.5: The mean error of the Gyroscope yaw angular velocity measurements

In conclusion, we found that

$$\begin{aligned} \mu_{g,k} &= 0.15\omega_{g,k} \\ \sigma_{g,k} &= 0.07|\omega_{g,k}| + 0.02 \end{aligned} \tag{3.7}$$

3.2.3 Absolute Measurement Unit

As explained previously, considering the pros and cons of different techniques for absolute position measurement, two approaches based on detection of landmarks stuck on the floor at known positions were chosen to adjust robot localization.

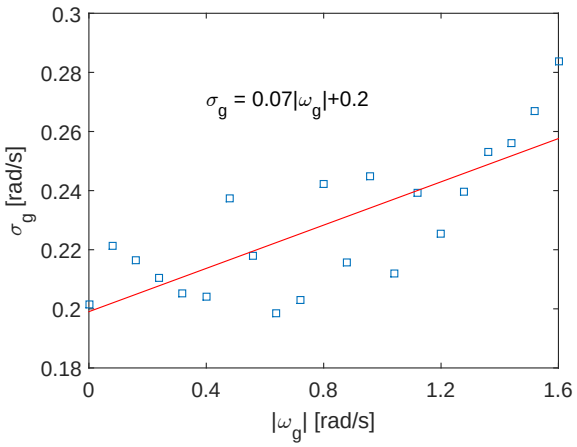


Figure 3.6: The standard deviation of the Gyroscope yaw angular velocity measurements

Passive RFID systems

Passive RFID tags have a very low cost and do not need any maintenance (e.g. battery replacement). Therefore, they are preferable for localization in large environments. The tags can be buried in carpets or stuck on the floor. They can be even installed under the mosaics or floor covers. Initially, we installed a high-frequency (HF) ISC.MR101-USB evaluation RFID reader made by Feig Electronics operating at 13.56 MHz inside the c-Walker. The reader is designed to detect passive tags compliant with Standards ISO 15693 (Vicinity Cards) and ISO 18000-3 (Smart Labels).

The performances of the passive RFID system for localization purposes were analyzed using the reader at hand. The main steps of the testing procedure are listed below.

- A tag was placed on the floor and the RFID reader was moved radially towards the tag and backwards in eight directions (i.e. north, north-east, east, south-east, south, south-west, west, north-west), while keeping the antenna parallel to ground.
- The reading range of the device in every direction was estimated by measuring repeatedly the distances at which the tag was detected or stopped being detected.
- After checking the omni-directional behavior of the reader, a histogram of the measured range values was built.

The results showed that the reader range has approximately a circular symmetry with radius $R = 15 \pm 1\ cm$. Hence, to a first approximation, anytime a tag is detected, the reader position measurement uncertainty is uniformly distributed over a circle centered in the tag and with radius R.

A variety of different experiments in various conditions were done for robot localization using passive RFID system [63, 64, 65]. The results show that the passive RFID

systems provide fast, reliable and accurate position information although the reading range is quite limited. However, when the tags are not read continuously, but sporadically with an unknown detection time, they cannot return any information about the absolute robot's heading. Hence, the heading estimation uncertainty accumulates causing a possible drift of position estimation as well. Thus, using another technology to measure absolute orientation is unavoidable [65]. Therefore, despite the valuable benefits of the passive RFID technology, we shifted to a complete vision-based solution able to measure the robot's absolute position and heading at the same time.

Vision systems based on floor mounted artificial landmarks

Although ad-hoc artificial landmarks can be designed for very fast image processing algorithms, they suffer the lack of generality. So, the whole image processing and detection algorithms should be redesigned as soon as landmarks change. If a standard coding technique is used, there is no need to design new artificial landmarks. For instance, a barcode is an optical label readable with various types of optical sensors. The label can contain different information such as a product code, a link or, in our application, a position. A QR code is a two-dimensional barcode invented in Japan in 1994. It consists of black squares inside a square grid with a white background (Figure 3.7). Using such squares, large amount of data can be stored. Also, they can be easily read by simple cameras. To this aim, three larger squares at the corners of a QR code, called *Finder Patterns* are used to locate the QR in the image, while one (or multiple) smaller squares near the fourth corner, called *alignment patterns*, are used to normalize the image for orientation and size [11]. The advantages of QR codes (mainly fast detectability and high capacity) have made them the most popular type of barcodes. As a consequence, their applications are rapidly increasing. They have been used as reference landmark points for pedestrian localization [8] and can be attached to the ceiling for mobile robots [106] and AAL applications [38]

A simple RGB-D camera can be used to detect the QR codes spread in the environment. The QR code can contain its absolute position or a simple number representing the position in a look-up table. Efficient open source libraries and algorithms are available to detect and read a QR code, such as Zbar library for C++[3]. Several algorithms have been developed to find the distance of the camera from the detected code as well as the orientation angle of the QR code with respect to the optical axis of the camera [32].

In this research, a cheap RGB-D camera (a simple USB PSeye RGB 640×480 webcam) viewing the floor is mounted in front of the robot. It is installed about 80 *cm* above the floor level with an orientation of about 60 degrees towards ground. In this way, the camera field of view ranges from 0.2 to 1.2 meters in front of the robot. The

[3] http://zbar.sourceforge.net/

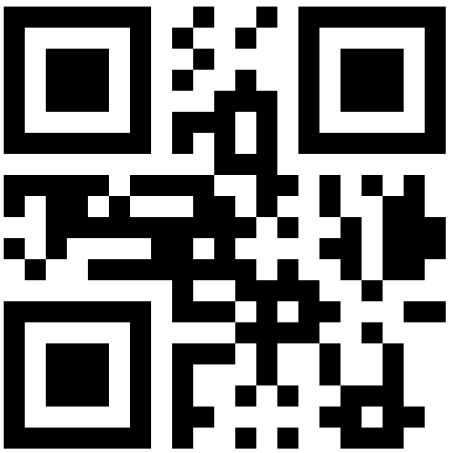
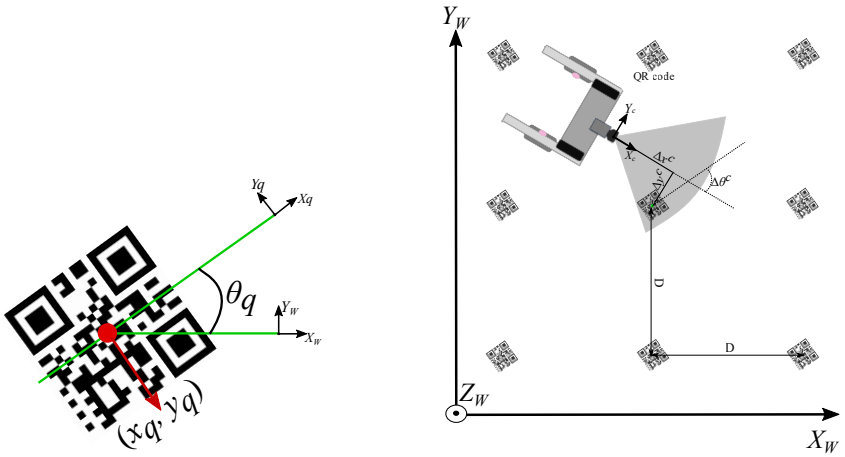


Figure 3.7: A sample QR code



(a) (b)

Figure 3.8: The camera observations: (a) A specifications of a QR code inside the reference frame, (b) the vision system measures the camera relative position and heading w.r.t the QR code in the camera frame

camera aperture angle is 30 degrees. Figure 3.8 shows how QR codes could be deployed in a given reference frame. In the proposed approach, a QR code stores only an integer number q, which is associated with the planar coordinates (x_q, y_q) and the direction θ_q of the QR code with respect to the x-axis of the given reference frame. This approach is very flexible, since the table associating each QR code number to (x_q, y_q, θ_q) can be easily changed and adapted to different environments, without reprinting the QR codes. Also, in this way just low-density numeric-only codes can be used. The choice of using low-density QR codes increases the ability to detect them at larger distances. The QR

codes used in the experiments are of version-1 QR codes with 21×21 black-and-white cells and a type L (i.e. low-level) Reed-Solomon error correction coding (ECC) is adopted. All QR codes were generated according to the ISO/IEC Standard 18004:2006 and printed on regular papers with a resolution of 600 dpi. A prominent specification of the QR codes is their size which should be chosen according to the maximum distance from the camera. As a rule of thumb, QR code size should be at least one order of magnitude smaller than the scanning range. Also, their size should be proportional to a data density factor given by the ratio between the number of columns (or rows) of the chosen QR code type (i.e. 21 in the case considered) and the number of columns (or rows) of a standard version-2 code (i.e. 25). Since the scanning range is about 1.2 m on average, after some tests, the QR codes size was set equal to 15×15 cm. QR code detection relies on the open-source Zbar library. QR code landmark recognition is instead implemented in C++ using the primitives of the OpenCV library. QR code detection and recognition is performed by the Intel Nuc Mini-PC. The distance between the camera image plane and the central point of the QR code is estimated in the camera local frame $\langle X_c, Y_c \rangle$ to determine Δx^c and Δy^c (see Figure 3.8-b). The image is then rotated to make the QR code coordinate frame $\langle X_q, Y_q \rangle$ parallel to the camera frame coordinates $\langle X_c, Y_c \rangle$. The rotation gives the heading difference $\Delta\theta^c$. Finally, the measurements returned by the camera and image processing system at time kT_s (if a QR code is detected at this time) are

$$\begin{aligned} \widehat{\Delta x}_k^c &= \Delta x_k^c + \zeta_{x,k}^c \\ \widehat{\Delta y}_k^c &= \Delta x_k^c + \zeta_{y,k}^c \\ \widehat{\Delta\theta}_k^c &= \Delta x_k^c + \zeta_{\theta,k}^c \end{aligned} \tag{3.8}$$

where $\zeta_{x,k}^c$, $\zeta_{y,k}^c$ and $\zeta_{\theta,k}^c$ are the uncertainties of Δx_k^c, Δy_k^c and $\Delta\theta_k^c$ measurements at time kT_s, respectively.

The vision system characterization

In order to estimate the probability density functions (PDFs) of uncertainty terms ζ_x^c, ζ_y^c and ζ_θ^c, the robot was repeatedly driven towards a target landmark in different directions and at different speeds till detecting the QR code. Every time, the robot was stopped as soon as the triple of values $[\widehat{\Delta x}^c, \widehat{\Delta y}^c, \widehat{\Delta\theta}^c]^T$ measured by the vision system was available. The actual values of Δx^c and Δy^c after halting the robot were measured with a yardstick and a square ruler (with a residual uncertainty of about ± 1 cm). The values of $\Delta\theta^c$ instead were estimated with a goniometer centered in the intersection point between the virtual optical axis of the camera and the reference direction of the QR code when $\theta_q = 0$.

Figures 3.9, 3.10, 3.11 show the PDFs of ζ_x^c, ζ_y^c and ζ_θ^c, respectively, estimated using about 100 experimental data. The black lines obtained from a nonlinear fitting of the underlying histograms correspond to a log-logistic, a triangular and a Gaussian distribution for ζ_x^c, ζ_y^c and ζ_θ^c, respectively. It is worth noticing that the latencies to extract the measures from the collected images are random and can be so large as 150 ms. As a result, the displayed distributions of ζ_x^c, ζ_y^c and ζ_θ^c are significantly affected by such latencies. On the contrary, the worst-case uncertainty values when the robot is stock-still are about one order of magnitude smaller than those shown in figures 3.9, 3.10, 3.11. Experimental evidence showed that ζ_x^c, ζ_y^c and ζ_θ^c are quite correlated when the robot is still. However, they tend to become uncorrelated when the robot is in motion probably because the random processing latencies affect ζ_x^c, ζ_y^c and ζ_θ^c independently.

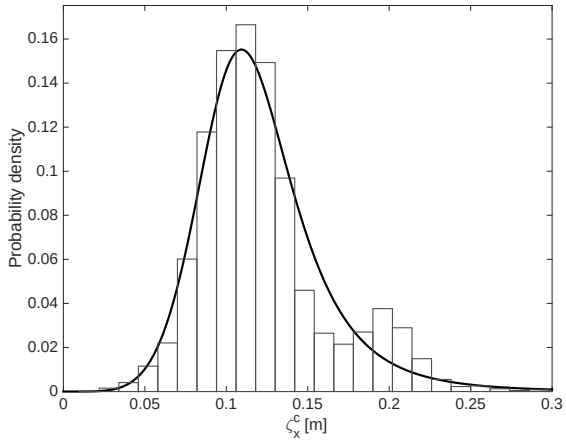


Figure 3.9: The PDF of the error corresponding to the measurement of Δx^c. The best fitting distribution is Log-logistic with scale parameter $\alpha_{x^c} \approx -2.15$ and shape parameter $\beta_{x^c} \approx 0.17$. The vision method computational burden effects the measurment significantly.

The difference between the distributions of ζ_x^c and ζ_y^c is mainly due to the adopted setup. Since in the experiments considered (as well as in typical scenarios) the camera field of view is maximum in the direction of its optical axis, the impact of image acquisition and processing latencies is much larger on Δx^c than on Δy^c. Also, Δx^c is always overestimated because the robot mainly moves forwards. As a result, the distribution of ζ_x^c is skewed with a mean value of about 12 cm, whereas ζ_y^c exhibits a zero mean because the probabilities of detecting a QR code located on the right or on the left of the optical axis are approximately the same. Similarly, the mean value of ζ_θ^c is zero, as the robot can be reasonably assumed to turn left and right with equal probability. Therefore, just the mean value of Δx^c has to be properly compensated (i.e. subtracted from $\widehat{\Delta x}^c$) prior to using the measured data in the localization algorithms.

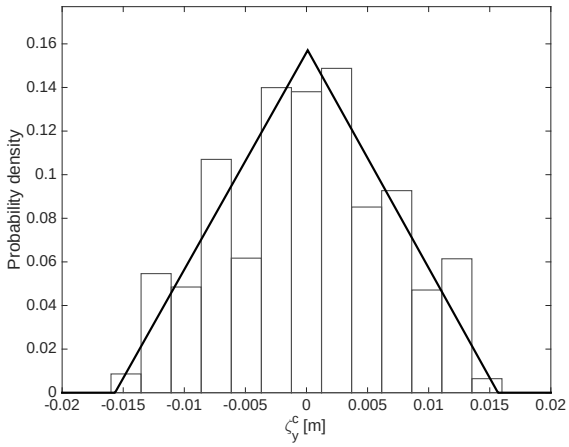


Figure 3.10: The PDF of the error corresponding to the measurement of Δy^c. The best fitting distribution is Triangular with lower limit $a_{y^c} \approx -0.015$, upper limit $b_{y^c} \approx 0.015$ and mode $c_{y^c} \approx 0$.

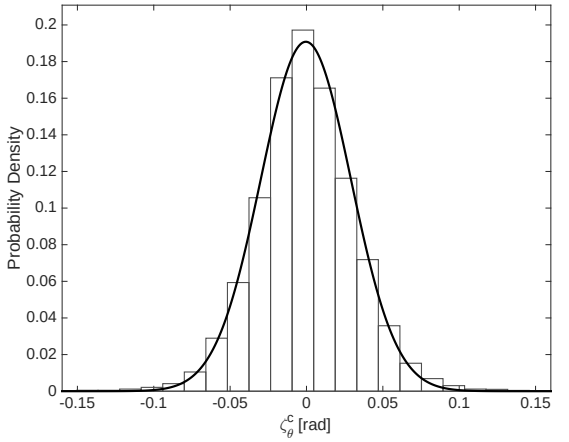


Figure 3.11: The PDF of the error corresponding to the measurment of $\Delta\theta^c$. The best fitting distribution is Gaussian with mean $\mu_{\theta^c} \approx 0$ and standard deviation $\sigma_{\theta^c} \approx 0.033$.

3.2.4 A summary of the proposed measurement techniques

In conclusion, the proposed localization system relies on odometry using two encoders, a gyroscope and absolute pose measurements given by a vision system detecting QR codes stuck on the floor. While Figure 3.12 shows an overview of the proposed sensing technologies, Table 3.2 summarizes the main characteristics of the sensors described in this chapter.

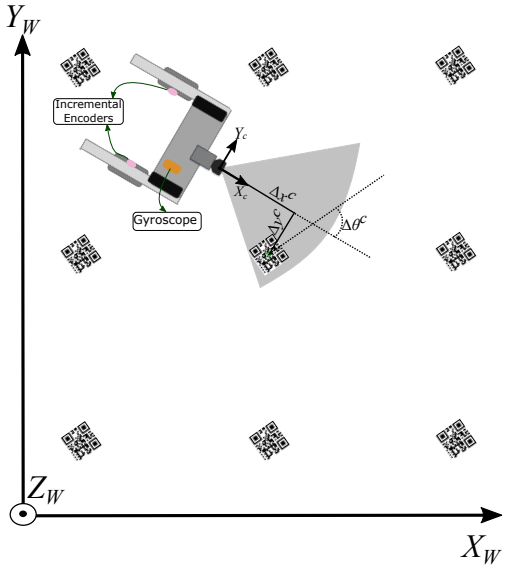


Figure 3.12: An overview of the total sensing technologies.

Table 3.2: Overview of all the sensors

Sensor	**Parameter**	**Noise Distribution**	**Noise properties**
Left odometer	$\Delta\Phi_l$	Gaussian	$\mu_{\phi_l} = 0.01\Delta\Phi_l$ rad $\sigma_{\Phi_l} = 0.002$ rad
Right odometer	$\Delta\Phi_r$	Gaussian	$\mu_{\phi_r} = 0.01\Delta\Phi_r$ rad $\sigma_{\Phi_r} = 0.002$ rad
Gyroscope	ω_g	Gaussian	$\mu_g = 0.15\omega$ rad/s $\sigma_g = 0.07\lvert\omega\rvert + 0.2$ rad/s
Camera	Δx^c	Loglogistic	$\alpha_{x^c} \approx -2.15$ $\beta_{x^c} \approx 0.17$
	Δy^c	Triangular	$a_{y^c} \approx -0.015$ $b_{y^c} \approx 0.015$
	$\Delta\theta^c$	Gaussian	$\mu_{\theta^c} = 0$ rad $\sigma_{\theta^c} = 0.033$ rad

Chapter 4

Multi-sensor Data Fusion Algorithms for Localization

In this chapter, some of the multi-sensor data fusion algorithms, introduced in Chapter 2 are applied and properly designed for robot position tracking using the sensing technologies described in Chapter 3. To this purpose, the first step is to define the proper state space model of the system and to find the state variables that could be observed using the available measurements. Therefore, a process model and a measurement model should be first defined.

4.1 Process model

The process model of mobile robots describes the system locomotion using the current states and inputs to the system. It usually depends on the motion dynamics or kinematics of the mobile agent and can be defined as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \boldsymbol{\nu}) \tag{4.1}$$

where $\mathbf{x}$ is the vector of states used in the localization system, $\mathbf{u}$ is the vector of system inputs and $\boldsymbol{\nu}$ represents the uncertainty of the states. Various types of state variables can be defined in a system depending on the localization technique. Some possible state variables for this purpose are [67]:

- *Position* (in 1, 2 or 3 dimensions)
- *Forward or angular velocity* (in 1, 2 or 3 dimensions)
- *Acceleration* (in 1, 2 or 3 dimensions)
- *Orientation* (in 1, 2 or 3 dimensions)

- *Position and/or orientation of reference points in the environment* (in 2 or 3 dimensions)
- *Sensors systematic biases and offsets*
- *Measurement or communication delays*

After determining the desired states, the process model should be defined appropriately. Although the nature of motion is continuous, it can be discretized with a sampling time T_s as follows

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\nu}_k) \tag{4.2}$$

where k denotes the *k-th* sample.

4.1.1 Unicycle Robots

Standard models of wheeled robots include unicycle, car-like and omni-directional models. Although the methods and technologies used and discussed in this dissertation are general and can be applied to different models, only the unicycle model will be used in the rest of this dissertation. The unicycle-like robots are indeed commonly adopted in service robotics and the unicycle model is the best one to describe the dynamics of c-Walker.

Unicycle-like robots are non-holonomic vehicles whose motion relies on two parallel wheels which are mounted on the sides of a center of mass. Their movement is path dependent and cannot have any kind of instantaneous lateral motion. A qualitative overview of an unicycle-like vehicle moving inside a local navigation reference frame $\langle W \rangle = \{X_w, Y_w, Z_w\}$ is shown in Figure 4.1.

The robot's generalized coordinates are given by $\mathbf{p} = [x, y, \theta]^T$, where (x, y) are the planar coordinates of the mid-point of the wheels axle and θ is the heading of the robot with respect to X_w. Point $\mathbf{p}$ which coincides with the origin of the robot reference frame [93] is referred to as robot reference point in this dissertation. The unicycle robot locomotion can be described by the following model [93]

$$\begin{aligned} \dot{\mathbf{p}} &= \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \\ \mathbf{o} &= \mathbf{h}(\mathbf{p}) \end{aligned} \tag{4.3}$$

where v and ω are the linear and angular velocities of the robot, $\mathbf{o}$ represent the system output, and $\mathbf{h}(\cdot)$ denotes a generic system output function. The inputs of (4.3), namely

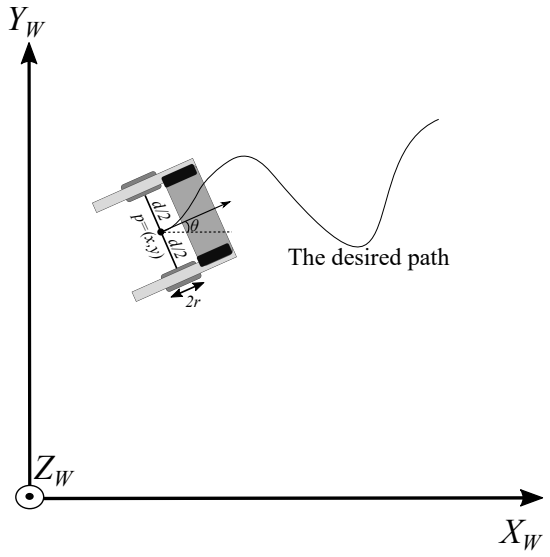


Figure 4.1: A unicycle-like vehicle

v and ω, are linearly related to the right and left wheels velocities by

$$\begin{aligned} v &= \frac{r}{2}(\omega_r + \omega_l), \\ \omega &= \frac{r}{d}(\omega_r - \omega_l), \end{aligned} \tag{4.4}$$

where r is the wheels radius, d is the length of the wheels axle, and ω_r, ω_l are the angular velocities of the right and left wheel, respectively.

4.2 The proposed system model

The wheels angular velocities, i.e. ω_r and ω_l, can be measured using the encoders data as follows

$$\begin{aligned} \omega_r &= \frac{\Delta\Phi_r}{T_s} \\ \omega_l &= \frac{\Delta\Phi_l}{T_s} \end{aligned} \tag{4.5}$$

Therefore, the encoders can be used to estimate the inputs v and ω of the system as well. However, the inputs estimated by the wheels displacement and encoders, are usually affected by some uncertainty due to noise, and wheels and system uncertainties. In particular, the major systematic contributions to angular velocity uncertainties are due to poor calibration of encoders, wheels radius and axle length tolerances, and mechanical

gearboxes imperfections. In general, the systematic contributions to angular and linear velocity should be added to the system model. As a result of the encoders characterization (see Chapter 3), some further experimental results and some references in the literature, e.g. [21], the systematic contributions can be approximately regarded as linear functions of the corresponding velocities. Thus, the augmented state space model based on (4.3) is

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\mu} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} (1+\mu)v\cos\theta \\ (1+\mu)v\sin\theta \\ (1+\delta)\omega \\ 0 \\ 0 \end{bmatrix} \tag{4.6}$$

where μ and δ are the relative systematic offsets affecting the values of v and ω respectively. Thus,

$$\dot{\mathbf{s}} = \mathbf{f}_w(\mathbf{s})\boldsymbol{\Omega}, \tag{4.7}$$

where

$$\mathbf{f}_w(\mathbf{s}) = \begin{bmatrix} \frac{r}{2}(1+\mu)\cos\theta & \frac{r}{2}(1+\mu)\cos\theta \\ \frac{r}{2}(1+\mu)\sin\theta & \frac{r}{2}(1+\mu)\sin\theta \\ \frac{r}{d}(1+\delta) & -\frac{r}{d}(1+\delta) \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

is the process function and $\boldsymbol{\Omega} = [\omega_r, \omega_l]^T$ is the input vector. Finally, if (4.7) is discretized with sampling time T_s, it can be rewritten as

$$\mathbf{s}_{k+1} = \mathbf{s}_k + T_s\mathbf{f}_w(\mathbf{s}_k)\boldsymbol{\Omega}_k = \mathbf{s}_k + \mathbf{f}_w(\mathbf{s}_k)\boldsymbol{\Delta\Phi}_k, = \mathbf{f}(\mathbf{s}_k, \boldsymbol{\Delta\Phi}_k) \tag{4.8}$$

where $\mathbf{s}_k$ is the state vector at time kT_s and $\boldsymbol{\Delta\Phi}_k = [\Delta\Phi_{r_k}, \Delta\Phi_{l_k}]^T$ is the vector containing the right and left wheels angular displacement between time kT_s and $(k+1)T_s$. Since the wheels displacements are measured by encoders, the real input is an estimation of the wheels displacement, i.e. $\boldsymbol{\Delta\Phi}_k = \widehat{\boldsymbol{\Delta\Phi}}_k + \boldsymbol{\varepsilon}_k^{\boldsymbol{\Phi}}$. Hence, the system model is rewritten as

$$\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \widehat{\boldsymbol{\Delta\Phi}}_k + \boldsymbol{\varepsilon}_k^{\boldsymbol{\Phi}}) \tag{4.9}$$

4.3 Measurement Model

The measurement model describes the sensor measurements as a function of the states. In fact, the state variables should be chosen properly such that they can explain the sensors'

observations. In general, the measurement model is defined as

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\zeta}) \tag{4.10}$$

where $\mathbf{y}$ is the vector of sensors measurements, $\mathbf{h}(.)$ is the function relating the sensor measurements to the state variables and $\boldsymbol{\zeta}$ represents the measurements uncertainty. The corresponding discretized measurement model is

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\zeta}_k) \tag{4.11}$$

4.3.1 Measurement model of the selected sensors

In the case study described in Chapter 3, the sensors chosen for the measurement model are the vision system measuring Δx^c, Δy^c and $\Delta\theta^c$, and the IMU estimating the heading angle, θ^g. In total, the measurement can be divided into two groups, i.e.

1. Measurements of relative position $\widehat{\Delta x}^c = \Delta x^c + \zeta_x^c$ and $\widehat{\Delta y}^c = \Delta y^c + \zeta_y^c$
2. Measurements of heading that can be obtained by merging $\widehat{\Delta\theta}^c = \Delta\theta^c + \zeta_\theta^c$ and $\hat{\theta}^g = \theta^g + \varepsilon^g$

Position measurement model

The vision system has to be able to measure the distances Δx^c and Δy^c between the camera and the point (x_q, y_q) associated with the center of any detected QR code. However, if the camera is not co-located in the robot reference point used to define the position of the robot (as it typically occurs in practice), the coordinates of the reference point are functions of Δx^c and Δy^c through a constant rigid transformation. If the position of the camera is fixed, this transformation can be computed once and for all by measuring the relative position of the camera with respect to the robot reference point. Therefore, given the detected QR code position $\mathbf{p}_q = [x_q, y_q]$, the transformation vector from robot reference point to the camera position $\boldsymbol{l}^c = [l_x^c, l_y^c]$ and the system states vector, the position measurement model can be defined as

$$\mathbf{o}_k^c = \mathbf{h}^c(\mathbf{s}_k) = \begin{bmatrix} (x_q - x_k)\cos\theta_k + (y_q - y_k)\sin\theta_k + l_x^c \\ -(x_q - x_k)\sin\theta_k + (y_q - y_k)\cos\theta_k + l_y^c \end{bmatrix}. \tag{4.12}$$

Moreover, since the elements of $\mathbf{o}_k^c$ are measured by the vision system, the corresponding measurement model becomes

$$\hat{\mathbf{o}}_k^c = \begin{bmatrix} \widehat{\Delta x}_k^c \\ \widehat{\Delta y}_k^c \end{bmatrix} = \mathbf{h}^c(\mathbf{s}_k) + \boldsymbol{\zeta}_k^c \tag{4.13}$$

where $\boldsymbol{\zeta}_k^c = [\zeta_{x,k}^c, \zeta_{y,k}^c]^T$ is the measurement uncertainty vector, as explained above.

Heading measurement model

Let $\mathbf{q} = [\theta^g, b^g]^T$ be a vector composed of the yaw angle θ^g and the relative angular velocity uncertainty offset b^g associated with the gyroscope-based platform. If b^g is assumed to be approximately constant and is not properly estimated and compensated, then the systematic error affecting the angular velocity observed by the gyroscope is proportional to ω. If the sampling period of the gyroscope is T_s, then the discretized heading measurement model is

$$\begin{aligned} \mathbf{q}_{k+1} &= \begin{bmatrix} \theta^g_{k+1} \\ b^g_{k+1} \end{bmatrix} = \begin{bmatrix} \theta^g_k + T^g_s(1+b^g_k)\omega^g_k \\ b^g_k \end{bmatrix} + \boldsymbol{\varepsilon}^g_k \\ o^\theta_k &= h^\theta(\theta^g_k) \end{aligned} \tag{4.14}$$

where $\boldsymbol{\varepsilon}^g$ denotes the noise introduced by the gyroscope-based platform and $h^\theta(\cdot)$ is the output function which depends on the yaw angle only, since b^g usually cannot be observed in practice. If the vision system is used to measure $\Delta\theta^c$, then it can be shown that $o^\theta_k = h^\theta(\theta^g_k) = \Delta\theta^c_k = \theta_q - \theta^g_k$. Therefore, the corresponding measurement equation becomes $\hat{o}^\theta_k = \widehat{\Delta}\theta^c_k = o^\theta_k + \zeta^c_k$, ζ^c_k being the heading measurement uncertainty due to the vision system at time kT_s. If the elements of $\boldsymbol{\varepsilon}^g$ and ζ^c are uncorrelated and normally distributed with zero mean, the values of $\mathbf{q}$ can be suboptimally estimated by an EKF based on (4.14). As known, an EKF relies on two iterative steps (namely, *prediction* and *update*) applied to the linearized dynamic [6]. In this case, the equations of the *prediction step* are

$$\begin{aligned} \hat{\mathbf{q}}_{k+1|k} &= \hat{\mathbf{q}}_{k|k} + \mathbf{g}(\hat{\mathbf{q}}_{k|k})\hat{\omega}_k, \\ Q_{k+1|k} &= A_k Q_{k|k} A^T_k + B_k \sigma^2_{\omega,k} B^T_k, \end{aligned} \tag{4.15}$$

where $\hat{\mathbf{q}}_k$ and $\hat{\mathbf{q}}_{k+1|k}$ denote the estimated and predicted state, respectively, $\mathbf{g}(\hat{\mathbf{q}}_k) = \begin{bmatrix} T^g_s(1+\hat{b}^g_k) \\ 0 \end{bmatrix}$ is the input function, $\hat{\omega}_k$ is the angular velocity value given by the gyroscope-based platform at time kT^g_s, $A_k = \begin{bmatrix} 1 & T^g_s\hat{\omega}_k \\ 0 & 1 \end{bmatrix}$ and $B_k = \begin{bmatrix} T^g_s(1+\hat{b}^g_k) \\ 0 \end{bmatrix}$ are the Jacobians of the process model (4.14) with respect to $\mathbf{q}$ and ω, respectively, computed at $(\hat{\mathbf{q}}_k, \hat{\omega}_k)$; $Q_{k|k}$ and $Q_{k+1|k}$ are the estimated and predicted state covariance matrices at times kT_s and $(k+1)T_s$, respectively. Finally, $\sigma^2_{\omega,k}$ is the variance of the gyroscope noise at time kT^g_s.

Using the same notation, the *update step* equations of the EKF, when a QR code is detected, are

$$\begin{aligned} K^\theta_{k+1} &= Q_{k+1|k}C^T\left(CQ_{k+1|k}C^T + \sigma^2_{\theta^c,k+1}\right)^{-1} \\ \hat{\mathbf{q}}_{k+1|k+1} &= \hat{\mathbf{q}}_{k+1|k} + K^g_{k+1}(\hat{o}^\theta_{k+1} - C\hat{\mathbf{q}}_{k+1|k}) \\ Q_{k+1|k+1} &= (I_2 - K^\theta_{k+1}C)Q_{k+1|k} \end{aligned} \tag{4.16}$$

where K^{θ}_{k+1} is the Kalman gain at time $(k+1)T_s$, $C = \begin{bmatrix} -1 & 0 \end{bmatrix}$ is the system output matrix and $\sigma^2_{\theta^c,k+1}$ is the variance of ζ^c_θ. Note that if no landmark is detected at time kT_s, no update is possible. Thus, the EKF works in open loop. In conclusion, the heading estimation θ^g_k is used as the heading measurement in the main system model.

The overall measurement model

According to the description reported above, we have an event-based measurement model that depends on which sensors data are available at each sampling time T_s. Indeed, the available measurements at time kT_s can be

$$\hat{\mathbf{o}}_k = \begin{cases} \begin{bmatrix} \widehat{\Delta x}^c_k \\ \widehat{\Delta y}^c_k \\ \hat{\theta}^g_k \end{bmatrix} & \text{QR detected} \\ \hat{\theta}^g_k & \text{Only IMU data available} \end{cases}$$

This results in the following output model

$$\hat{\mathbf{o}}_k = \mathbf{h}(\mathbf{s}_k) + \boldsymbol{\zeta}_k = \begin{cases} \begin{bmatrix} \mathbf{h}^c(\mathbf{s}_k) \\ \theta_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{\zeta}^c_k \\ \zeta^g_{\theta,k} \end{bmatrix} & \text{QR detected} \\ \theta_k + \zeta^g_{\theta,k} & \text{Only IMU data available} \end{cases} \tag{4.17}$$

where $\zeta^g_{\theta,k}$ is the heading estimation uncertainty of $\hat{\theta}^g_k$. In particular, the variance of $\zeta^g_{\theta,k}$ should coincide with the $(1,1)$ element of the covariance matrix Q_k defined in (4.16).

4.4 Estimators for Sensors Data Fusion

As introduced in Chapter 2, depending on the application, different dynamic estimators can be used. System (4.6) is non-linear, the process noise is non-additive and as shown in section 3.2, some of the measurements uncertainty contributions are not Gaussian. Hence, four different types of estimators are analyzed and compared in the following, i.e.

1. The Extended Kalman Filter (EKF)
2. The Unscented Kalman Filter (UKF)
3. The Particle filter (PF) based on Sequential Importance Sampling
4. The Extended H_∞ Filter (EHF)

In the rest of this chapter each technique is introduced, described and applied to the problem at hand. In all methods, the estimation is based on the discretized model (4.8).

Even if the estimation can be performed in one step, it is more convenient to use the usual *prediction-update* two-step approach. In all cases, the prediction step is based on the same process model (4.9) while the update step relies on the measurement model (4.17). This two-step approach is specially required when the measurements are event-based, like in the problem of this dissertation. The aim of the prediction step is to find an optimal a priori estimation of the current states given the previous states, i.e. $\mathbf{s}_{k|k-1}$. When any measurement is available, the update step will modify the predicted state variables, thus returning the updated state at the current time, i.e. $\mathbf{s}_{k|k}$. When dealing with intermittent, multi-rate or at all every event-based measurements, the two-step estimation approach allows the system to work in open-loop, i.e. using only the prediction step when no measurement is available. In this situation, the predicted estimates are also accepted as the a-posteriori information, i.e. $\mathbf{s}_{k|k} = \mathbf{s}_{k|k-1}$

4.4.1 Extended Kalman Filter (EKF)

The extended Kalman filter is in fact a nonlinear version of the Kalman filters which linearizes the system about the current state mean and estimated covariance.

Initialization

Usually, the mean and covariance of the initial state should be known. If improper initial values are used, filter may diverge. This does not mean that the initial state, i.e. the robot initial position, must be known; but it means that we should have knowledge about how accurate the initial state is. So,

$$\begin{aligned} \hat{\mathbf{s}}_{0|0} &= \hat{\mathbf{s}}_0 = E[\mathbf{s}_0] \\ P_{0|0} &= P_0 = E[(\mathbf{s}_0 - \hat{\mathbf{s}}_0)(\mathbf{s}_0 - \hat{\mathbf{s}}_0)^T] \end{aligned} \tag{4.18}$$

Prediction

In the EKF, the state is predicted using the previous state values, the previous sample inputs and the nonlinear process model, i.e.

$$\hat{\mathbf{s}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{s}}_{k|k}, \widehat{\mathbf{\Delta\Phi}}_k), \tag{4.19}$$

The predicted covariance results from a Taylor-based linearization, i.e.

$$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k E_k G_k^T. \tag{4.20}$$

where F_k is the Jacobian of the state space model with respect to $\mathbf{s}_{k|k}$ at $[\hat{\mathbf{s}}_{k|k},\ \widehat{\mathbf{\Delta\Phi}}_k]$, i.e.,

$$\begin{aligned} F_k &= \frac{\partial[\mathbf{f}(\hat{\mathbf{s}}_{k|k}, \widehat{\mathbf{\Delta\Phi}}_k)]}{\partial \hat{\mathbf{s}}_{k|k}}\Big|_{[\hat{\mathbf{s}}_{k|k},\ \widehat{\mathbf{\Delta\Phi}}_k]} = \\ &= \begin{bmatrix} 1 & 0 & F_k^{13} & F_k^{14} & 0 \\ 0 & 1 & F_k^{23} & F_k^{24} & 0 \\ 0 & 0 & 1 & 0 & F_k^{35} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{4.21}$$

with

$$\begin{aligned} F_k^{13} &= -\frac{r}{2}(1+\hat{\mu}_k)\left(\widehat{\Delta\Phi}_{r_k} + \widehat{\Delta\Phi}_{l_k}\right)\sin\hat{\theta}_k \\ F_k^{14} &= \frac{r}{2}\left(\widehat{\Delta\Phi}_{r_k} + \widehat{\Delta\Phi}_{l_k}\right)\cos\hat{\theta}_k \\ F_k^{23} &= \frac{r}{2}(1+\hat{\mu}_k)\left(\widehat{\Delta\Phi}_{r_k} + \widehat{\Delta\Phi}_{l_k}\right)\cos\hat{\theta}_k \\ F_k^{24} &= \frac{r}{2}\left(\widehat{\Delta\Phi}_{r_k} + \widehat{\Delta\Phi}_{l_k}\right)\sin\hat{\theta}_k \\ F_k^{35} &= \frac{r}{d}\left(\widehat{\Delta\Phi}_{r_k} - \widehat{\Delta\Phi}_{l_k}\right) \end{aligned}$$

Moreover, G_k is the Jacobian of the state space model with respect to $\widehat{\Delta\Phi}_{r_k}$ and $\widehat{\Delta\Phi}_{l_k}$ computed at $[\hat{\mathbf{s}}_{k|k},\ \widehat{\mathbf{\Delta\Phi}}_k]$, i.e.,

$$G_k = \frac{\partial[\mathbf{f}(\hat{\mathbf{s}}_{k|k}, \widehat{\mathbf{\Delta\Phi}}_k)]}{\partial \mathbf{\Delta\Phi}_k}\Big|_{[\hat{\mathbf{s}}_{k|k},\ \widehat{\mathbf{\Delta\Phi}}_k]} = \begin{bmatrix} \frac{r}{2}\cos\hat{\theta}_k & \frac{r}{2}\cos\hat{\theta}_k \\ \frac{r}{2}\sin\hat{\theta}_k & \frac{r}{2}\sin\hat{\theta}_k \\ \frac{r}{d} & -\frac{r}{d} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.22}$$

E_k is the covariance matrix associated with the encoders uncertainty $\boldsymbol{\varepsilon}_k$. Since the encoders are nominally identical, but independent, their uncertainty contributions are weakly correlated in practice. Therefore, E_k is a 2×2 diagonal matrix with $\sigma^2_{\Phi_r,k}$ and $\sigma^2_{\Phi_l,k}$ (defined in (3.4)) on the main diagonal.

Update

Whenever, measurement data are available, a Kalman gain can be computed and used to update the predicted estimates of the states. The Kalman gain is given by [6]

$$K_{k+1} = P_{k+1|k}H_{k+1}^T\left(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right)^{-1}, \tag{4.23}$$

where H_{k+1} is the Jacobian of the output $\mathbf{h}(\mathbf{s})$ and depends on the set of available measures. The Kalman gain and the measurement data record $\hat{\mathbf{o}}_{k+1}$ are finally used to update the system state estimate and the corresponding covariance matrix as follows

$$\begin{aligned} \hat{\mathbf{s}}_{k+1|k+1} &= \hat{\mathbf{s}}_{k+1|k} + K_{k+1}\left(\mathbf{o}_{k+1} - h(\hat{\mathbf{s}}_{k+1|k})\right), \\ P_{k+1|k+1} &= (I - K_{k+1}H_{k+1})P_{k+1|k}. \end{aligned} \tag{4.24}$$

4.4.2 Unscented Kalman Filter (UKF)

The sigma-point Kalman filters are other members of the nonlinear Kalman filtering family which provide a linearization solution using weighted statistical linear regression. They linearize a nonlinear function of a random variable through a linear regression between some specific points, called sigma points, drawn from the prior distribution of the random variable. This results in two major advantages comparing with the EKF [100], i.e.

1. The actual uncertainty of the system states are taken into account while EKF disregards the probabilistic spread of the noise during the linearization.

2. The first order Taylor series linearization of EKF results in first order accuracy of the propagated mean and covariance while the accuracy in sigma-point filters is of the second order

The unscented Kalman filter (UKF) is a kind of sigma-point filter in which the sigma points are chosen in such a way that they can capture the most important statistical properties of the prior state.

The first step in designing the UKF is to specify the number of sigma points N_s. In the normal method of UKF, $N_s = 2L + 1$ where L is the dimension of system state. In the presence of nonlinear or non-additive model uncertainties, some of them should be regarded as additional state variables [102]. In the case considered, the augmented state for the UKF would be $\mathbf{s^a} = [\mathbf{s}, \boldsymbol{\varepsilon}, \boldsymbol{\zeta}]^T$ where $\boldsymbol{\varepsilon}$ and $\boldsymbol{\zeta}$ are the process and measurement noise vectors respectively. According to the model proposed in this dissertation, only the system uncertainty is non-additive while the measurement noise is linear and additive. Hence, the augmented states can be only variables associated to the encoders noise. i.e.,

$$\mathbf{s}_k^a = \begin{bmatrix} \mathbf{s}_k \\ \varepsilon_k^{\Phi} \end{bmatrix} = \begin{bmatrix} x_k & y_k & \theta_k & \mu_k & \delta_k & \varepsilon_k^{\Phi_r} & \varepsilon_k^{\Phi_l} \end{bmatrix}^T \tag{4.25}$$

Hence, in this system $L = 7$, $N_s = 2L + 1 = 15$ and $\mathcal{X}^a$ is the matrix of N_s sigma vectors:

$$\mathcal{X}^a = \left[\mathcal{X}_0^a, \mathcal{X}_{i,i=1\ldots 2L}^a\right] \tag{4.26}$$

Initialization

The proper initial augmented state is defined as

$$\begin{aligned}
&\hat{\mathbf{s}}_0 = E[\mathbf{s}_0] \\
&P_0 = E[(\mathbf{s}_0 - \hat{\mathbf{s}}_0)(\mathbf{s}_0 - \hat{\mathbf{s}}_0)^T] \\
&\hat{\mathbf{s}}_0^a = E[\mathbf{s}_0^a] = \begin{bmatrix} \hat{\mathbf{s}}_0 \\ 0 \\ 0 \end{bmatrix} \\
&P_0^a = E[(\mathbf{s}_0^a - \hat{\mathbf{s}}_0^a)(\mathbf{s}_0^a - \hat{\mathbf{s}}_0^a)^T] = \begin{bmatrix} \mathbf{P}_0 & 0_{5\times 2} \\ 0_{2\times 5} & \mathbf{E} \end{bmatrix}
\end{aligned} \tag{4.27}$$

where $0_{m\times n}$ is a zero matrix of size $m \times n$. Moreover, each sigma point affects the estimation with a predefined weight as follows

$$\begin{aligned}
&W_0^m = \lambda/(L+\lambda) \\
&W_0^c = \lambda/(L+\lambda) + 1 - \alpha^2 + \beta \\
&W_i^m = W_i^c = 1/[2(L+\lambda)] \quad for\ i = 1:2L
\end{aligned} \tag{4.28}$$

where $\lambda = \alpha^2(L+\kappa) - L$ is the UKF scaling parameter which should be chosen a priori. Parameter α can be used to obtain the sigma points spread around mean estimate $E[\mathbf{s}]$ and usually set to a small positive value, κ is a scale factor to be chosen (usually set to zero) and β is used to incorporate prior knowledge of the distribution of $\mathbf{s}$. W^m is used for the state estimation while W^c is the weight vector for the covariance estimation.

Sigma point calculation

The state variables for N_s sigma points should be computed at every sample time kT_s

$$\begin{aligned}
&\mathcal{X}_k^a = \begin{bmatrix} \mathcal{X}_{k,0}^a & \mathcal{X}_{k,i}^a & ... & \mathcal{X}_{k,2L+1}^a \end{bmatrix} = \begin{bmatrix} \mathcal{X}_k^s \\ \mathcal{X}_k^{\varepsilon} \end{bmatrix} \\
&\mathcal{X}_{k,0}^a = \hat{\mathbf{s}}_k^a \\
&\mathcal{X}_{k,i}^a = \hat{\mathbf{s}}_k^a + (\sqrt{(L+\lambda)P_k^a}) \qquad for \quad i = 1, ..., L \\
&\mathcal{X}_{k,i}^a = \hat{\mathbf{s}}_k^a - (\sqrt{(L+\lambda)P_k^a}) \qquad for \quad i = L+1, ..., 2L+1
\end{aligned} \tag{4.29}$$

The sigma points are then used to find a suboptimal Gaussian propagation of the state variables.

Prediction

Each sigma point is first passed through the system model to predict the states of the sigma points.

$$\mathcal{X}^s_{k+1|k,i} = \mathbf{f}(\mathcal{X}^s_{k,i}, \widehat{\mathbf{\Delta\Phi}}_k, \mathcal{X}^{\boldsymbol{\varepsilon}}_{k,i}) \qquad for \quad i = 0, ..., 2L \tag{4.30}$$

Finally, the predicted state and covariance are determined as

$$\hat{\mathbf{s}}_{k+1|k} = \sum_{i=0}^{2L} W_i^m \mathcal{X}^s_{k+1|k,i} \tag{4.31}$$

$$P_{k+1|k} = (\sum_{i=0}^{2L} W_i^c [\mathcal{X}^s_{k+1|k,i} - \hat{\mathbf{s}}_{k+1|k}][\mathcal{X}^s_{k+1|k,i} - \hat{\mathbf{s}}_{k+1|k}]^T) \tag{4.32}$$

Update

Each sigma point is passed through the measurement function to achieve the sigma point observation matrix $\mathcal{Z}$

$$\mathcal{Z}_{k+1,i} = h(\mathcal{X}^+_{k,i}) \tag{4.33}$$

From the observation matrix, the output can be estimated

$$\hat{\mathbf{z}}_{k+1} = \sum_{i=0}^{2L} W_i^m \mathcal{Z}_{k+1,i} \tag{4.34}$$

Using (4.33) and (4.34) the Kalman gain can be computed

$$\begin{aligned} P_{zz} &= \sum_{0}^{2L} W_i^c [\mathcal{Z}_{k+1,i} - \hat{\mathbf{z}}_{k+1}][\mathcal{Z}_{k+1,i} - \hat{\mathbf{z}}_{k+1}]^T \\ P_{xz} &= \sum_{0}^{2L} W_i^c [\mathcal{X}_i - \hat{\mathbf{s}}_{k+1|k}][\mathcal{Z}_{k+1,i} - \hat{\mathbf{z}}_{k+1}]^T \\ K_{k+1} &= P_{xz}(P_{zz} + R_k)^{-1} \end{aligned} \tag{4.35}$$

At last, the Kalman filtering updated state and covariance are given by

$$\hat{\mathbf{s}}_{k+1|k+1} = \hat{\mathbf{s}}_{k+1|k} + K_{k+1}(\mathbf{o}_{k+1} - \hat{\mathbf{z}}_{k+1}) \tag{4.36}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1}(P_{zz} + R_{k+1})K^T_{k+1} \tag{4.37}$$

4.4.3 Particle Filter (PF)

The Particle filters are suboptimal Bayesian filters which provide a point mass representation of the states estimation probability density using sequential Monte Carlo approaches [84, 2]. After developing in 1990s [40, 26], the PFs have rapidly become proper alternatives to the Kalman filters for nonlinear and non-Gaussian systems. A large variety of different PF algorithms have been introduced so far. Among them, the Sequential Importance Resampling (SIR) technique is used in this dissertation since it is a suitable standard method with less computational cost than other well-known techniques, although it has a much higher computational burden than Kalman filters. Like other estimators, due to the intermittency of measurements, the technique should be defined in a two-level predict-update way.

Initialization

The main parameter to be defined in the initial step is the number of particles N_p. The larger the number of particles, the better the estimation probability density becomes and the more the computational burden grows. A PF estimates the state of each particles. Hence, an initial state vector is defined for every particle depending on the a-priori knowledge of the agent's initial position. The estimated state at each sample is the weighted average of the estimated particles.

$$\begin{aligned} \mathbf{s}_0^i &= E[\mathbf{s}_0^i] \quad , i \in \mathbb{Z} \ , i \in [1, N_p] \\ \mathbf{w}_0^i &= \frac{1}{N_p} \\ \hat{\mathbf{s}}_0 &= \sum_{i=1}^{N_p} \mathbf{W}_0^i \mathbf{s}_0^i \end{aligned} \tag{4.38}$$

where $\mathbf{w}_0^i$ is the weight associated with particle number i.

Prediction

Starting from sample $k = 1$, the prediction is simply done by propagating each particle using the system model.

$$\mathbf{s}_{k+1}^i = f(\mathbf{s}_{k|k}^i, \widehat{\mathbf{\Delta\Phi}}_k, \boldsymbol{\varepsilon}_k^{\Phi}) \tag{4.39}$$

Update

Once the measurements from sensors are collected, the particle weights are modified by computing the measurements and system PDFs, i.e .

$$\tilde{\mathbf{w}}^i_{k+1} = \mathbf{w}^i_k + \frac{p(\mathbf{o}_{k+1}|\mathbf{s}^i_{k+1})p(\mathbf{s}_{k+1}|\mathbf{s}^i_k)}{q(\mathbf{s}_{k+1}|\mathbf{s}^i_k, o_{k+1})} \tag{4.40}$$

where q is the measurement *importance (or proposal) density* which should be chosen appropriately. Despite some drawbacks, the most common approach is to use the transition prior probability distribution as the importance function, i.e. $q(\mathbf{s}_{k+1}|\mathbf{s}^i_k, o_{k+1}) = p(\mathbf{s}_{k+1}|\mathbf{s}^i_k)$. This simplifies the recursive weight computation to

$$\tilde{\mathbf{w}}^i_{k+1} = \mathbf{w}^i_k + p(\mathbf{o}_{k+1}|\mathbf{s}^i_{k+1}) \tag{4.41}$$

The computed weight should be normalized to be used in a weighted average, i.e.

$$\mathbf{w}^i_{k+1} = \frac{\tilde{\mathbf{w}}^i_{k+1}}{\sum_{i=1}^{N_p} \tilde{\mathbf{w}}^i_{k+1}} \tag{4.42}$$

Thus, the state estimation can be updated as follows:

$$\hat{\mathbf{s}}_{k+1|k+1} = \sum_{i=1}^{N_p} \mathbf{w}^i_{k+1}\mathbf{s}^i_{k+1} \tag{4.43}$$

Note that the covariance matrix $P_{k+1|k+1}$ can be determined by computing the variance of each state variable from all particles. Furthermore, in the systems with event-based measurements, the weights are unchanged if no measurement is available at time kT_s, i.e. in such situation, $\mathbf{w}^i_{k+1} = \mathbf{w}^i_k$.

Sample depletion and resampling

Since the weight associated with each particle at any time kT_s depends on both its value in the previous samples and the current value, the weight of some particles would continuously decrease and approach to zero. Because of this problem, which is called *sample depletion* or *degeneracy*, after some iterations only few (or even sometimes one) particle will remain and contribute to state estimation. Some solutions to this problem have been discussed in the literature. One of the solutions which is usually used in every kind of particle filtering technique is resampling.

In resampling, the particles with negligible weights are substituted by some new particles with higher weights. This can partially solve the depletion problem but kills some of information of the measurements PDFs and the Monte Carlo random sampling uncertainty increases. Therefore, the resampling should not be applied at every sample but

only when depletion is probable. To this aim, the number of particles whose associated weight is not negligible (called effective number of particles) are found at every sampling time [84], i.e.

$$N_{eff} = \frac{N_p}{1 + N_p^2 \cdot \mathrm{var}(\mathbf{w}_{k+1})} \tag{4.44}$$

A threshold for the minimum number of effective particles (N_{th}) is chosen as a scale factor of the total particles number,

$$N_{th} = \beta N_p,\ \beta \in (0, 1)$$

and the resampling is applied only if $N_{eff} < N_{th}$. The resampling process aims at removing samples with very low importance and booting those with high importance. The *Select with Replacement* technique have been used in many applications so far, specially for mobile robot localization [83]. It is simple, effective and solves most of the sample depletion problems. But, on the other hand, its computational burden is pretty large due to the need for random number generation and sorting. A complete overview of some effective resampling algorithms can be found in [56].

Nonetheless, resampling cannot guarantee a complete solution of the depletion problem. Usually some other parameters need to be selected properly and/or modified depending on the system and application. A major parameter which should be selected appropriately rather than the resampling strategy, is the proposal distribution. If the measurement is very accurate, the actual noise distribution may lead to very small probabilities which increases the probability that the particles are depleted. To avoid this, the following ad-hoc modifications are sometimes recommended and applied in practice:

- The domain of the distribution chosen for the measurements ($p(\mathbf{o}_{k+1}|\mathbf{s}^i_{k+1})$ in (4.41)) should be $\mathbb{R}$. This prevents generating zero probabilities.
- Some references, like [40], recommend to use larger system and measurement noise models than the real sensor uncertainty models. This solution, which is called *jittering*, increases the propagation area and hence, decreases the potential number of depleted particles in every sampling time.

4.4.4 Extended $\mathbf{H}_\infty$ filter

The H_∞ filters provide a worst case optimal estimation approach to improve robustness to unmodelled noise and dynamics. They can robustly estimate the system states in the lack of statistical knowledge on the uncertainties model [61, 43]. Like the Kalman filters, for nonlinear systems the estimation is done sub-optimally using extended or unscented

approaches. However, the goal is to estimate a linear combination of the state at time kT_s, i.e.

$$\mathbf{y}_k = L_k \mathbf{s}_k \tag{4.45}$$

where L_k should be chosen appropriately.

As explained in [95], H_∞ filters can be obtained following a game theory approach. In particular, they are conceived to minimize a scalar cost function given by the ratio between the total weighted estimation error of a linear function of the desired state and a term depending on the initial estimation error as well as the model uncertainty and the measurement disturbances. In general, both the individual state estimation errors and the various uncertainty contributions can be weighted differently, depending on the nature of the problem at hand. In the robotic system of this dissertation, the objective function is

$$J_k \triangleq \frac{\sum_{j=0}^{k} \|L_j \boldsymbol{s}_j - L_j \hat{\boldsymbol{s}}_j\|_{I_j^l}^2}{\|\mathbf{s}_0 - \hat{\mathbf{s}}_0\|_{\tilde{P}_0^{-1}}^2 + \sum_{j=0}^{k} \|\boldsymbol{\varepsilon}_{\Phi,j}\|_{\tilde{E}_j^{-1}}^2 + \sum_{j=0}^{k} \|\boldsymbol{\zeta}_j\|_{\tilde{R}_j^{-1}}^2}, \tag{4.46}$$

where where I_j^l is an identity matrix whose dimension is equal to the number of rows of L_j, operator $\|\mathbf{v}\|_M^2 \triangleq \mathbf{v}^T M \mathbf{v}$ (with $\mathbf{v}$ and M being a generic vector and square matrix, respectively), the three weighting matrices at the denominator of (4.46), i.e. $\tilde{P}_0^{-1}$, $\tilde{E}_j^{-1}$, $\tilde{R}_j^{-1}$, offer various potential degrees of freedom for H_∞ filter design, the only constraint being that they must be symmetric and positive definite.

The cost function (4.46) should be properly minimized for any k. Unfortunately, the direct minimization of J_k generally is not a tractable problem. Nonetheless, a suboptimal solution can be obtained by finding the values of $\hat{\mathbf{s}}_k$ such that $J < \gamma^2$, with γ being a specified bound. This relaxed problem can be solved by using an array algorithm [43] and, in particular, the technique proposed in [57]. Following this approach, the EHF implementation can be split into a *prediction step* and an *update step*, similarly to an EKF.

Initialization

The initial state can be defined as in the EKF, i.e. based on (4.18). However, the EHF allows more flexibility in the choice of the initial state.

Prediction

As a result of all the discussions for EHF, the equations of the *prediction step* become the same as (4.19) and (4.20), with the only difference that $\tilde{E}_k$ replaces E_k. i.e.

$$\begin{aligned} \hat{\mathbf{s}}_{k+1|k} &= \mathbf{f}(\hat{\mathbf{s}}_{k|k}, \widehat{\boldsymbol{\Delta\Phi}}_k) \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + G_k \tilde{E}_k G_k^T. \end{aligned} \tag{4.47}$$

Update

The equations of the *update step* instead are [57]

$$\begin{aligned} \hat{\mathbf{s}}_{k+1|k+1} &= \hat{\mathbf{s}}_{k+1|k} + K_{k+1}\left(\mathbf{o}_{k+1} - \mathbf{h}(\hat{\mathbf{s}}_{k+1|k})\right), \\ P_{k+1|k+1} &= \left(I_5 - P_{k+1|k}\begin{bmatrix} H_{k+1}^T & L_{k+1}^T \end{bmatrix} U_{k+1}^{-1} \begin{bmatrix} H_{k+1} \\ L_{k+1} \end{bmatrix}\right) P_{k+1|k}, \end{aligned} \tag{4.48}$$

where H_{k+1} is the Jacobian of the output function $\mathbf{h}(\cdot)$ in (4.17) computed at $\hat{\mathbf{s}}_{k+1|k}$,

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + \tilde{R}_{k+1}\right)^{-1} \tag{4.49}$$

and

$$U_{k+1} = \begin{bmatrix} \tilde{R}_{k+1} & 0 \\ 0 & -\gamma_{k+1}^2 I_{k+1}^l \end{bmatrix} + \begin{bmatrix} H_{k+1} \\ L_{k+1} \end{bmatrix} P_{k+1|k} \begin{bmatrix} H_{k+1}^T & L_{k+1}^T \end{bmatrix}, \tag{4.50}$$

Notice that, using this formulation, the implementation of the proposed EHF is similar to an EKF, although the computational burden of the former estimator is a bit larger.

The EHF design consists of several parameters to be chosen appropriately. It can be shown that if the weighting matrices are equal to the corresponding EKF covariances, $L_k = I_5$ and $\gamma \to \infty$, then EHF and EKF tend to coincide. However, by selecting appropriate parameters the EHF results can be more robust and/or accurate than the EKF.

Selection of L_k

The EHF may become unstable if the state is not observable. In other words, the H_∞ minmax technique can be applied to the observable states. Hence, in the current system with intermittent and event-based measurements, the output linear function L_k defined in 4.45 should be changed in order to extract only the observable states at time kT_s. Note that this does not mean that we do not estimate the other states but means that the minimax filter is applied only to the observable quantities. In particular, due to the fact

that when only gyroscope data is available, just θ is observable and when a QR code is detected, x, y, and θ are observable, L_k is defined as

$$L_k = \begin{cases} \begin{bmatrix} I_3 & 0_{3\times 2} \end{bmatrix} & \text{QR detected} \\ \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \text{Only IMU data available} \end{cases} \tag{4.51}$$

Selection of the weight matrices

The weighting matrices $\tilde{P}_0$, $\tilde{E}_j$, $\tilde{R}_j$ can influence the cost function and hence, estimation accuracy. If some a-priori knowledge is available, $\tilde{P}_0$, $\tilde{E}_j$, $\tilde{R}_j$ should keep into consideration the relevance of different uncertainty contributions in vectors $\mathbf{s}_0 - \hat{\mathbf{s}}_0$, $\boldsymbol{\varepsilon}_{\Phi,j}$ and $\boldsymbol{\zeta}_j$, respectively. If these uncertainty contributions are white and normally distributed with a zero mean, then $\tilde{P}_0$, $\tilde{E}_j$, $\tilde{R}_j$ should coincide with the respective covariance matrices P_0, E_j and R_j defined for the EKF [95]. On the contrary, if such assumptions do not hold, the weighting matrices offer various degrees of freedom for EHF design. Indeed no univocal criteria for the selection of matrix parameters usually exist, besides the fact that $\tilde{P}_0$, $\tilde{E}_j$ and $\tilde{R}_j$ must be symmetric and positive definite. Moreover, if some a-priori knowledge is available, $\tilde{E}_j$ and $\tilde{R}_j$ can leverage the relative importance of the process noises by selecting different weights, i.e.

$$\tilde{E}_k = \alpha_{E,k}^2 E_k \tag{4.52}$$

E_k being the odometry covariance matrix, $\alpha_E = \text{diag}(\alpha_{\Phi^r,k}, \alpha_{\Phi^r,k})$ being a scale factor matrix, and

$$\tilde{R}_k = \alpha_{R,k}^2 R_k \tag{4.53}$$

where

$$\alpha_{R,k} = \begin{cases} \text{diag}(\alpha_x, \alpha_y, \alpha_\theta) & \text{QR detected} \\ \alpha_\theta & \text{Only IMU data available} \end{cases} \tag{4.54}$$

Selection of γ

The γ parameter of the EHF needs to be chosen with care. If $\gamma_k \to +\infty$, EHF behavior tend to become the same as EKF. On the contrary, the smaller γ_k, the more the EHF achieves optimality in a minimax sense. Unfortunately, if γ_k is too strict, matrix P_k in (4.48) might be no longer positive definite [28]. As a result, the solution of the relaxed optimization problem $J_k < \gamma_k^2$ might not exist, and the estimated trajectories could suddenly diverge. If the system is always observable, this problem can be addressed by changing γ_k adaptively, so that $P_{k|k}$ (as well as its inverse) is positive definite for any k.

In particular, by applying the matrix inversion lemma to (4.48), the condition of positive definiteness of $P_{k|k}^{-1}$ can be expressed as

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + H_k^T \tilde{R}_k^{-1} H_k - L_k^T \gamma_k^{-2} L_k > 0. \tag{4.55}$$

Hence, it is shown in [91] that P_k^{-1} is positive definite for all

$$\gamma_k^2 > \overline{\lambda}(L_k^T L_k (P_{k|k-1}^{-1} + H_k^T \tilde{R}_k^{-1} H_k)^{-1}) \tag{4.56}$$

where the function $\overline{\lambda}(M)$ returns the maximum eigenvalue of matrix M. Unfortunately, in the problem at hand, state variables x_k and y_k are not always observable, unless a QR code is detected at every sampling time. Because of such a lack of observability (which actually depends on the density of the QR codes in the room), the estimated trajectory might occasionally diverge, even if condition (4.56) holds and even if θ_k is observed. To avoid such divergence when only data from IMU is available (namely when $L_k = [0,\ 0,\ 1,\ 0,\ 0]$ and $\tilde{R}_k = \alpha_\theta^2 \sigma_{\theta,k}^2$ is a scalar), γ_k should be set larger than in (4.56), even if this choice may lead to suboptimal results. In particular, since $P_{k|k-1}$ is positive definite and $H_k^T \tilde{R}_k^{-1} H_k$ is positive semi-definite, it can be easily shown that if

$$P_{k|k-1}^{-1} - L_k^T \gamma_k^{-2} L_k > 0 \ \text{ and } \ H_k^T \tilde{R}_k^{-1} H_k - L_k^T \gamma_k^{-2} L_k \geq 0 \tag{4.57}$$

then $P_{k|k}^{-1}$ in (4.55) is certainly positive definite. Thus, in this case the values of γ_k result respectively from

$$\gamma_k^2 > \overline{\lambda}(L_k^T L_k P_{k|k-1}) \qquad \text{and} \qquad \gamma_k^2 \geq \tilde{R}_k = \alpha_\theta^2 \sigma_{\theta,k}^2. \tag{4.58}$$

Observe that the rightmost expression of (4.58) results from the fact that $L_k = H_k$ and, consequently, $H_k^T H_k (\tilde{R}_k^{-1} - \gamma_k^{-2}) \geq 0$. Ultimately, in order to avoid finite escape time phenomena while keeping accuracy as high as possible, γ_k can be computed adaptively by merging conditions (4.56) and (4.58) as follows:

$$\gamma_k^2 = \begin{cases} \xi \overline{\lambda} \left(L_k^T L_k (P_{k|k-1}^{-1} + H_k^T \tilde{R}_k^{-1} H_k)^{-1} \right) & \text{QR detected} \\ \max\{ \xi \overline{\lambda} \left(L_k^T L_k P_{k|k-1} \right)^{\frac{1}{2}}, \alpha_\theta^2 \sigma_{\theta,k}^2 \} & \text{QR not detected} \end{cases} \tag{4.59}$$

where ξ is a constant coefficient to be set slightly larger than 1.

Chapter 5

Experimental Results

To evaluate and compare the performance of position tracking methods described in Chapter 4, multiple experiments were conducted using the c-Walker robotic platform. A square-patterned grid of QR codes was regularly deployed on the floor of a 10 $m \times 15$ m room. The QR codes were placed at various distances D (i.e. $1, 2, 3, 4$ m) from each other. The c-Walker was guided randomly in the room over different paths. Figure 5.1 shows the c-Walker moving randomly in the room.

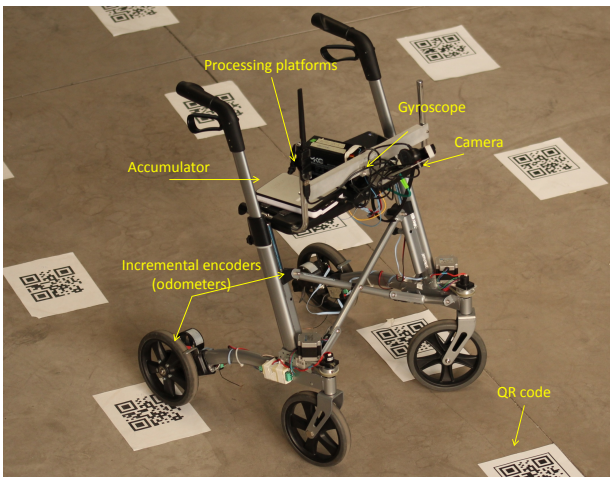


Figure 5.1: The c-Walker prototype moving randomly in an open space

To reconstruct the actual position of the robot (i.e. the ground truth), a SICK-S300 Expert laser scanner was placed in the origin of the navigation frame $\langle W \rangle$ (i.e. in one corner of the room) to measure the coordinates of the user along each route in real-time. The laser scanner has an angular resolution of 0.5°, a maximum scanning angle of up to 270° (but this was limited to 90° to increase the scanning rate) and a maximum reading range of about 30 m. The laser scanner can detect any object in the environment

and measure the position in the chosen reference frame. The robot can be identified by removing all the objects with static position after each experiment. To reduce the laser scanner estimation uncertainty the user was always trying to stand on the c-Walker reference point and the scanner was put on top of a 2-m-high shelf to detect just a hat wore by the user instead of the whole c-Walker. In this way, the cluster of points collected from the scanner at a given time was quite concentrated around robot reference point and finding the position of the centroid of the cluster was simpler. The heading ground-truth was instead reconstructed by differentiating points of consecutive positions. such data were interpolated with *spline* algorithm to smooth the trajectories.

Around 45 experiments of 180 *seconds* each were done in the room for each value of D and the sensors and ground-truth data were saved for further analysis. Also, the results of the localization algorithms running on the c-Walker and the ground truth were aligned in time. The system , encoders and IMU sampling times were set to $T_s = T_s^g = 4\ ms$ with the camera capturing 10 frames per second. The time intervals between two subsequent QR code detections are random, as they depend on the robot's trajectory. An example of a 180 second trajectory with a distance between QR codes of $D = 2\ m$ is shown in Figure 5.2. This trajectory is called the "*sample trajectory*" in the rest of this chapter.

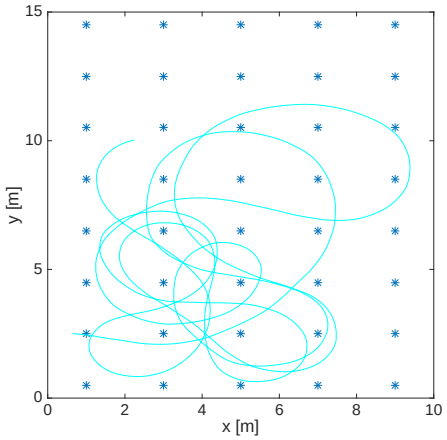


Figure 5.2: An 180 s random trajectory. The QR codes are deployed in a square grid with distance $D = 2\ m$. The Asterisk markers show the QR codes positions.

The accuracy of EKF, UKF, PF and EHF was tested starting from both known and unknown initial positions and headings. In both cases, the initial value of linear and angular velocity systematic contributions (i.e. μ and δ) was set to zero supposing that their uncertainty is uncorrelated and has zero-mean normal distribution with $\varepsilon_{\mu,0} \sim \mathcal{N}(0, 0.003)$ and $\varepsilon_{\delta,0} \sim \mathcal{N}(0, 0.0016)$

5.1 Extended Kalman filter

To use the EKF, the variance of noise of each sensor should be known. Using the same data used in sensor characterization, the uncertainty associated with Δx^c measurement (ζ_x^c), has mean $\mu_{\zeta_x^c} \approx 0.11\ m$ (which can be compensated) and standard deviation $\sigma_{\zeta_x^c} \approx 0.04\ m$. Similarly, the uncertainty associated with Δy^c measurement (ζ_y^c) is $\mu_{\zeta_y^c} \approx 0\ m$ and $\sigma_{\zeta_y^c} \approx 0.007\ m$. On the other hand, the uncertainty associated with the heading estimation can be supposed to be zero-mean and normally distributed with variance equal to element (1,1) of matrix Q_k in (4.16) described in Chapter 4. Hence,

$$R_k = \begin{cases} diag(\sigma^2_{\zeta_x^c}, \sigma^2_{\zeta_y^c}, \sigma^2_{\zeta_\theta^g}) & \text{QR detected} \\ \sigma^2_{\zeta_\theta^g} & \text{Only IMU data available} \end{cases} \tag{5.1}$$

Figure 5.3 shows the EKF estimation of the sample trajectory while Figure 5.4 represents the corresponding estimation error. It can be seen that the estimated trajectory completely follows the actual one. However, due to sporadic and event-based updates from the QR codes, the estimation error accumulates before a QR code detects and declined significantly when a new QQ code is detected. This fact leads to some spikes in the estimated trajectory which can be easily seen in Figure 5.3.

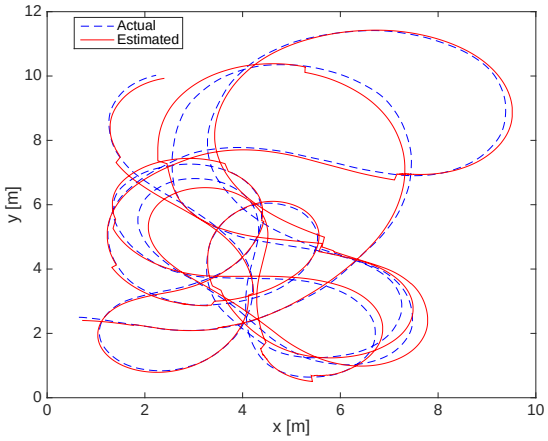


Figure 5.3: The EKF estimation of the sample trajectory

All the experiments with different QR distances ($D \in 1, 2, 3, 4\ m$) have been analyzed and the estimation error has been computed for each trajectory. The root mean square (RMS) position and heading estimation errors for all the experiments over 180 seconds are shown in Figure 5.5. Note that, in this chapter, the term *position estimation error* refers to the euclidean distance between the actual and estimated positions.

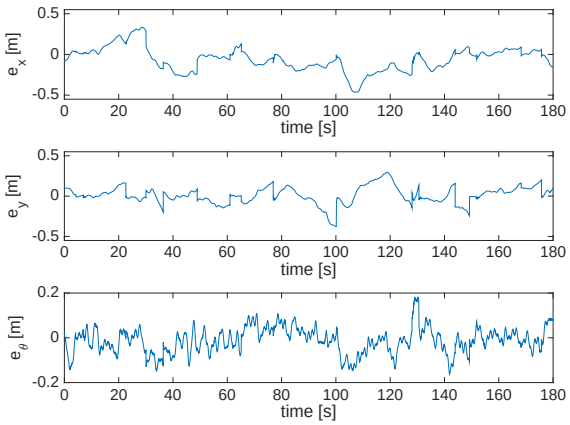
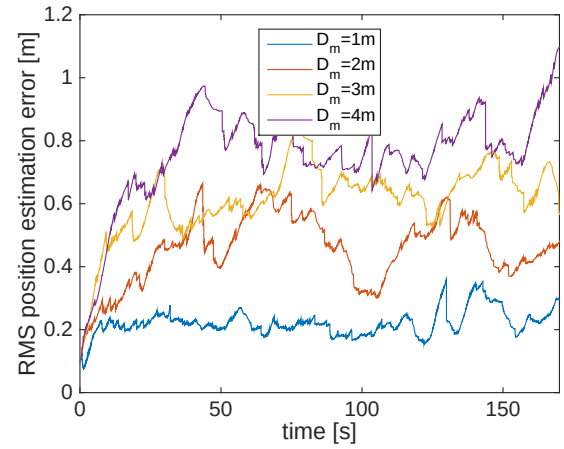
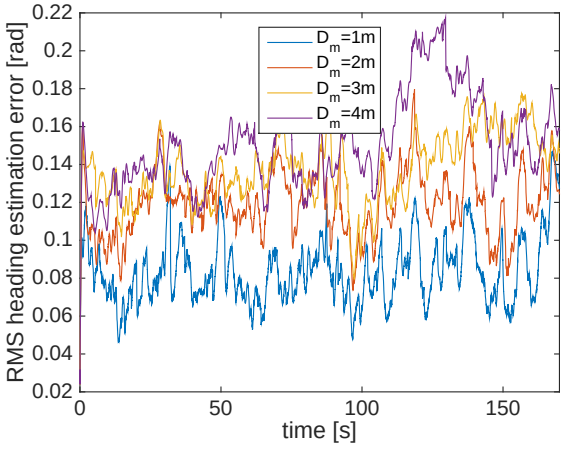


Figure 5.4: Estimation errors associated with the EKF for the sample trajectory.



(a)



(b)

Figure 5.5: The RMS pose estimation error for $D_M = 1\ m, 2\ m, 3\ m, 4\ m$ using EKF

5.2 Unscented Kalman filter

The Unscented Kalman filter (EKF) described in section 4.4.2 can also be used for the position tracking of the wheeled robots. The same measurement uncertainty model as EKF was used and UKF scale parameters were set to: $\alpha = 0.01$, $\beta = 2$, $\kappa = 0$.

The sample trajectory estimated by UKF and the corresponding estimation errors are shown in Figure 5.6 and Figure 5.7 respectively.

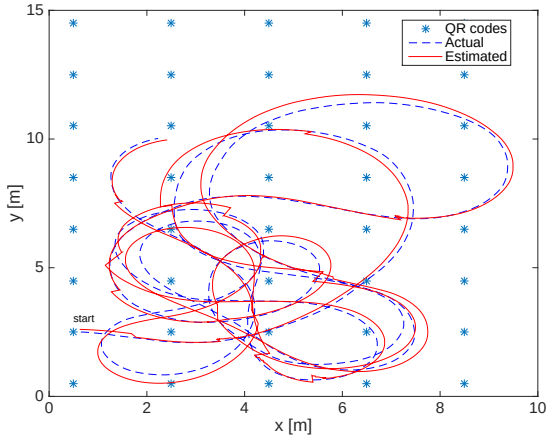


Figure 5.6: The UKF estimation of the sample trajectory

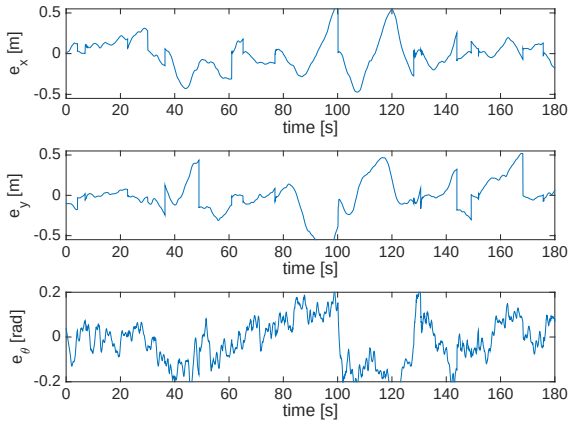


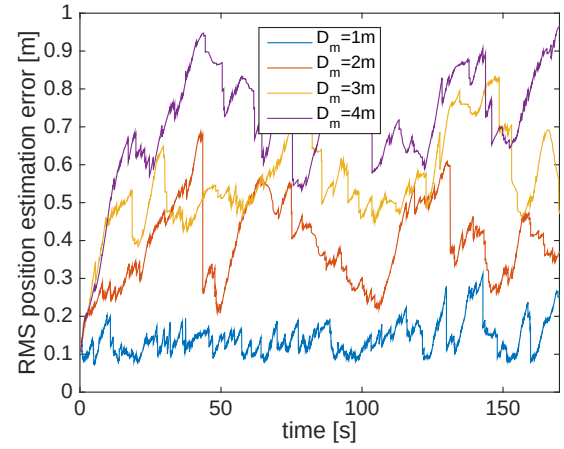
Figure 5.7: Estimation errors associated with the UKF for the sample trajectory.

Besides, Figure 5.8 shows the RMS position and heading estimation errors over time.

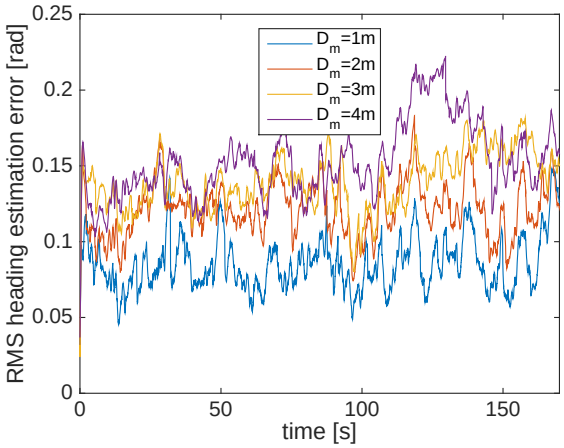
5.3 Particle filter

The PF parameters used in the experiments are summarized in the following:

1. Number of the particles $N_p = 1000$.



(a)



(b)

Figure 5.8: The RMS pose estimation error for $D_M = 1\ m, 2\ m, 3\ m, 4\ m$ using UKF

2. The threshold for the effective sample size was set to 75%, i.e. in this case $N_{eff} = 750$.

3. All the sensors measurement noises were supposed to have zero-mean Gaussian distribution with standard deviation 10 times larger than the ones used in EKF and UKF.

Even if all the particles should have probability density functions larger than zero, some zero probabilities are computed when the probability is less than the embedded platform precision. Hence, the probabilities computed as zero were set equal to smallest system precision, i.e. 2^{-52} in double precision floating point format. Figure 5.9 shows the EKF estimation of the sample trajectory. The corresponding estimation errors are shown in Figure 5.10.

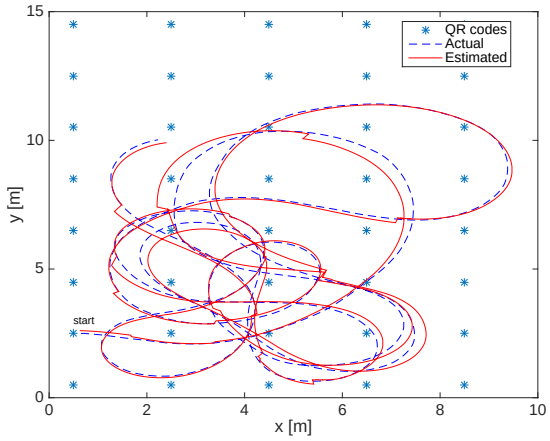


Figure 5.9: The PF estimation of the sample trajectory

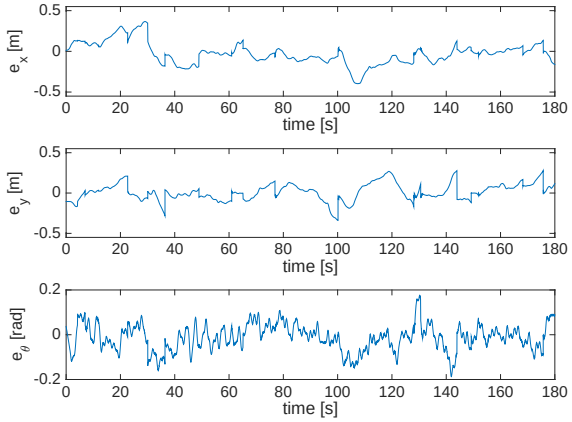


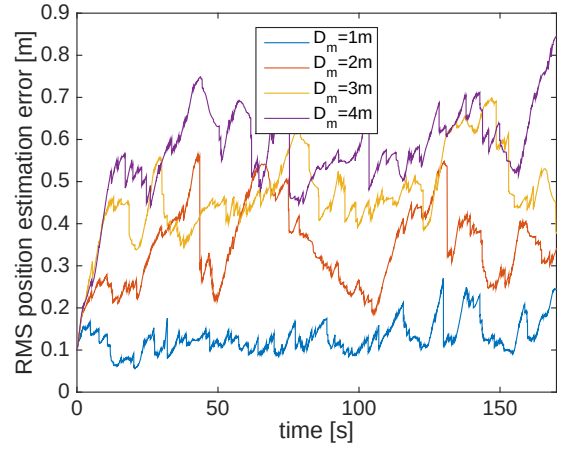
Figure 5.10: Estimation errors associated with the PF for the sample trajectory.

Finally, Figure 5.11 shows the RMSE of the position and heading estimation for all the experiments over time.

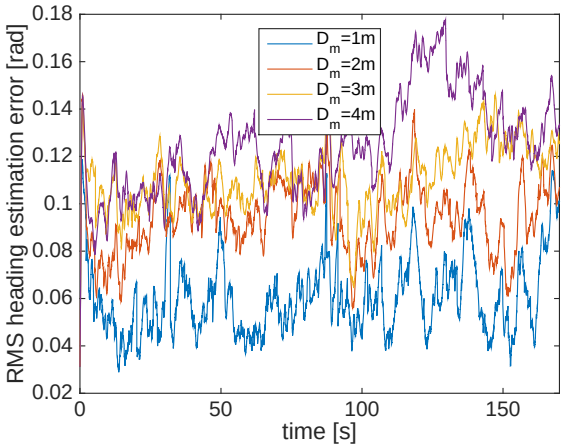
5.4 Extended H_∞ filter

Although the Extended H_∞ filter (EHF) has the advantage of not being influenced by the noise distribution, it requires to set a variety of parameters. Fortunately, the method is adaptive and ensures a robust behavior in different operating conditions. Nonetheless the weight matrices $\tilde{E}_k$ and $\tilde{R}_k$ are not required to coincide with the measurement covariance matrices and can be tuned to optimize the accuracy. Since the noises associated with the encoders are white, the best $\tilde{E}_k$ will coincide with the covariance [95]. Therefore, $\alpha_{E,k} = I_2$ and $\tilde{E}_k = E_k = diag(\sigma^2_{\phi_r}, \sigma^2_{\phi_l})$.

The problem of finding the best α_R, described in (4.54), is intractable analytically. So it was addressed through simulations based on experimental data. To this end, the



(a)



(b)

Figure 5.11: The RMS pose estimation error for $D_M = 1\ m, 2\ m, 3\ m, 4\ m$ using PF

EHF was repeatedly applied off-line to the same set of raw sensor data collected by the c-Walker, for different values of α_x, α_y and α_θ. From the comparison between the estimated results and those obtained with the laser scanner, it was observed empirically that:

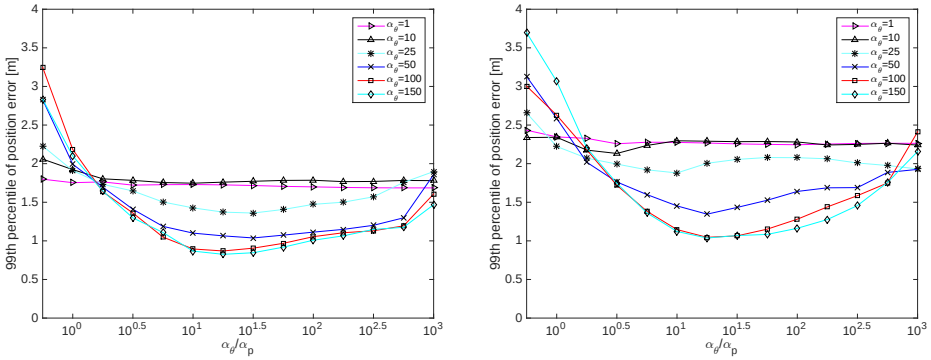
1. The best results are obtained when $\alpha_x \approx \alpha_y$. Thus, a single common coefficient (called α_p in the following) can be replaced to both α_x and α_y in (4.54).

2. Two suboptimal values of coefficients α_p and α_θ can be derived heuristically by finding the pair which minimizes the 99th percentile (computed over all available paths) of the Euclidean distance between actual and estimated positions.

The choice of using the 99th percentile of the position error as a performance index is dictated by the fact that the EHF is optimal in a minimax sense. So the EHF parameters

should be chosen to minimize the worst-case errors, while filtering possible outliers that could make the estimated maxima excessively noisy. Figure 5.12 shows the 99th percentile curves for different values of α_p and α_θ, and for (a) $D=2\ m$ and (b) $D=3\ m$, respectively.

To better highlight the configuration providing the best accuracy, the 99th percentiles of the position errors have been plotted as a function of the ratio α_θ/α_p for several values of α_θ. Most of curves in Figure 5.12 exhibit a minimum which does not change significantly when $\alpha_\theta \geq 100$. Thus, the EHF estimator accuracy is maximized for $\alpha_\theta \approx 100$ and $\alpha_p = \alpha_\theta/10^{1.25} \approx 5.6$.



(a) $D=2\ m$ (b) $D=3\ m$

Figure 5.12: 99th percentiles of the position estimation error for various values of α_θ as a function of ratio α_θ/α_p

The estimation of the sample trajectory based on EHF and the corresponding errors are shown in figures 5.13 and 5.14 respectively.

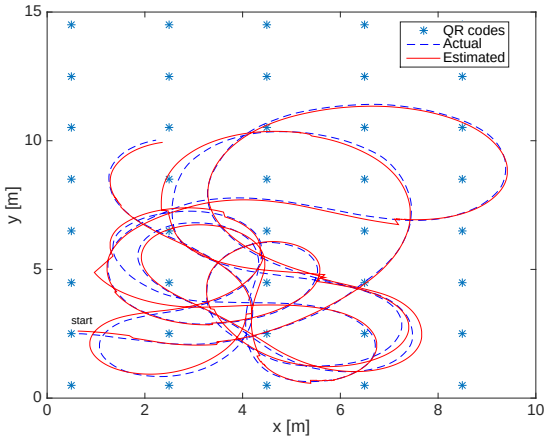


Figure 5.13: The EHF estimation of the sample trajectory

Besides, Figure 5.15 shows the RMSE of the position and heading estimation for all the experiments over time.

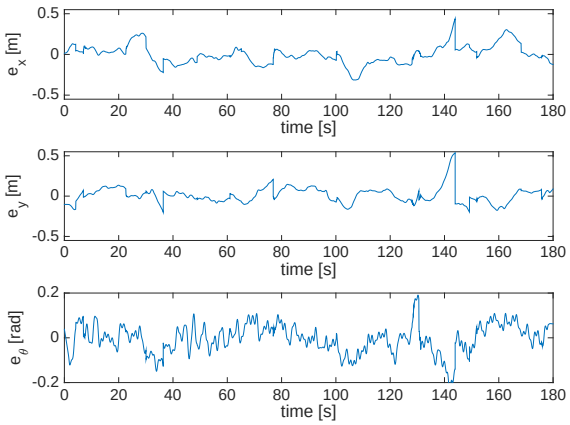
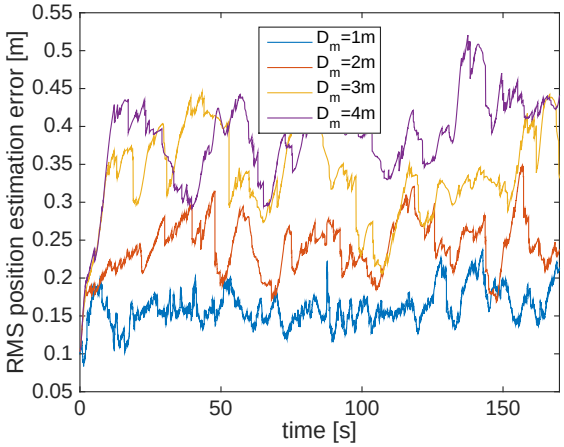
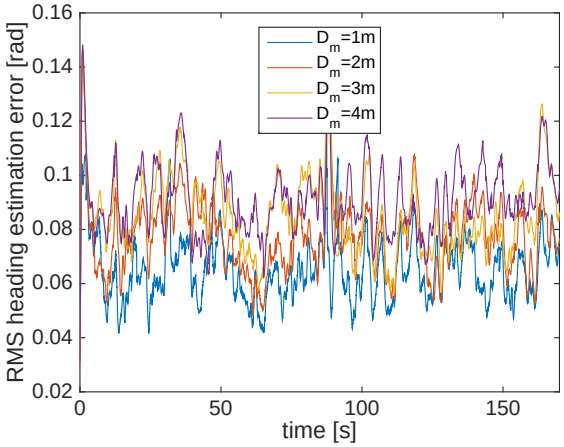


Figure 5.14: Estimation errors associated with the EHF for the sample trajectory.



(a)



(b)

Figure 5.15: The RMS pose estimation error for $D_M = 1\ m, 2\ m, 3\ m, 4\ m$ using EHF

5.4.1 A note on the H_∞ parameter selection

The choice of different parameters selection adds some degrees of freedom in designing H_∞ filters. However, it should be noticed that although proper parameter selection can result in more optimal estimation, inappropriate parameter selection can lead the system to instability, divergence or finite escape problems or sometimes decrease the robustness. The prominent parameter to avoid the finite escape is γ which was set as explained in Chapter 4. The weighting parameters instead affect the accuracy, robustness and reliability. They can robustly define a weight between measurements and provide much better estimation accuracy if selected appropriately. However, in some situations, they can affect the filter performance and robustness. For instance, in the case that the agent's location is lost, due to various reasons such as unknown or improper initialization or the so-called kidnapping problem, this can affect the estimation convergence since the system relies on the measurements less than the Kalman filters using the suboptimal weights calculated heuristically. In short, the smaller the values of α_p, α_θ and especially γ, the more robust the minimax filter becomes. On the contrary, the larger γ, the more stable the system is. Hence, depending on the application, such problems may need to be studied and addressed.

5.5 Filter comparison

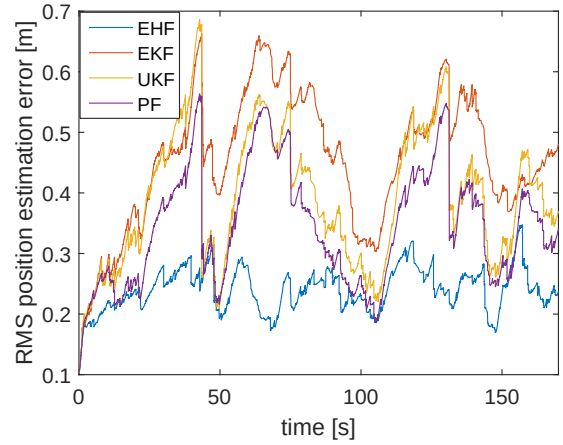
After clarifying each filtering method and showing the results, in this section, all the four estimators are compared and their advantages and disadvantages are discussed.

5.5.1 Accuracy

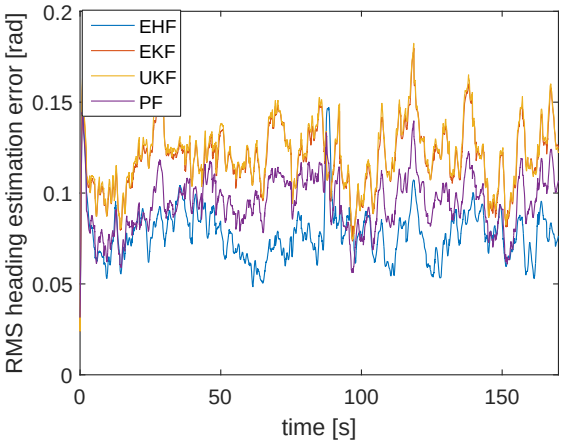
Figure 5.16 shows the RMSE of the position and heading estimation using all the filters for $D = 2\ m$. To provide a better clarification and accuracy comparison, the total RMS and 99th percentile estimation error values of x, y and θ have been computed in the steady state conditions. It was observed that to reach the steady state at least five updates from QR code detection are needed. The results are shown in Table 5.1. It can be clearly recognized that the proposed EHF outperforms other estimators, especially in the worst-case, although the PF also provides more accurate estimation compared with EKF and UKF.

5.5.2 Computational burden

Table 5.2 shows the average computational times of the proposed recursive filters running in the c-Walker embedded platform. It can be seen that in terms of average computation



(a)



(b)

Figure 5.16: RMS position and heading estimation errors of all four proposed filters with $D = 2\ m$

Table 5.1: RMS and 99th percentiles of the x, y and θ estimation errors (in steady state) obtained with various algorithms using regular grids of QR codes of a different granularity.

		D = 1 m				**D = 2 m**				**D = 3 m**				**D = 4 m**			
		EHF	**EKF**	**UKF**	**PF**	**EHF**	**EKF**	**UKF**	**PF**	**EHF**	**EKF**	**UKF**	**PF**	**EHF**	**EKF**	**UKF**	**PF**
x	RMSE [cm]	10	20	15	10	15	30	30	20	20	40	35	25	20	45	40	30
	99th Perc. [cm]	35	60	55	40	65	120	110	85	80	135	130	90	80	145	135	110
y	RMSE [cm]	10	20	15	10	15	30	30	20	20	40	35	30	20	45	40	35
	99th Perc. [cm]	35	60	50	40	50	115	105	85	70	130	125	95	80	145	140	120
θ	RMSE [crad]	5	10	10	5	10	15	15	10	10	15	15	10	10	15	15	10
	99th Perc. [crad]	20	30	30	25	25	40	40	35	30	45	45	35	25	50	50	45

time, the EHF is just slightly slower than the EKF, but it is much faster than both the UKF and PF, especially when a QR code is used to update the state of the system. Moreover, the distribution curves of computation times show that the EKF and the EHF can always return a result in real-time (i.e. within $T_s = 4$ ms), whereas in the worst

case the UKF and, above all, the PF may occasionally miss the deadline. This is evident especially in the PF case, since its average computation time is always larger than 4 ms when a QR code is detected.

Table 5.2: Computation time of the proposed EHF, EKF, UKF and PF for $D = 1$ m.

	EHF	**EKF**	**UKF**	**PF**
Computation time (no QR code update) [ms]	0.09	0.06	0.25	0.80
Computation time (with QR code update) [ms]	0.26	0.16	0.60	6.69

5.5.3 Unknown initial pose and convergence time

The results shown till now have been all obtained with known initial positions, due to the available ground-truth. In practice, the initial position is often unknown and the estimator should be able to converge to the actual ones after some updates.

The convergence time was evaluated with unknown initial position for $D = 1\ m$. In particular, the initial pose was defined randomly while the initial variances were set to proper large values. The RMS estimation error of all the trajectories over time using known and unknown initial pose were determined and compared. The convergence time was estimated as the time at which the estimation difference goes and remains below 10 cm for x and y and 10 $crad$ for θ. The results are summarized in Table 5.3.

Table 5.3: Convergence time of the EHF, the EKF, the UKF and the PF for $D = 1\ m$.

	EHF	**EKF**	**UKF**	**PF**
Convergence time [s]	18	9	2	11

As can be seen in Table 5.3, the UKF provides the shortest as its RMS estimation error rapidly approaches to the steady state value.

In the case of completely wrong initialization or case of kidnapping, the proposed EKF, UKF and EHF could overcome the problem in most situations (although sometimes after a significant time), with EHF showing the most robustness. However, the PF was not able to converge the estimation in this case. This is due to the natural behavior of the SIR particle filtering. To solve this problem, other PF methods such as marginal particle filters may be used. However, such methods have more computational cost than the SIR.

Observe that the proposed EHF converges more slowly than the others. As described in section 5.4, this can be due to the chosen weight matrices. To show this, the same approach described above was applied to the EHF in the case that the scale factors α_p and α_θ are set to one. The result decreased the convergence time from 18 to around 9 seconds.

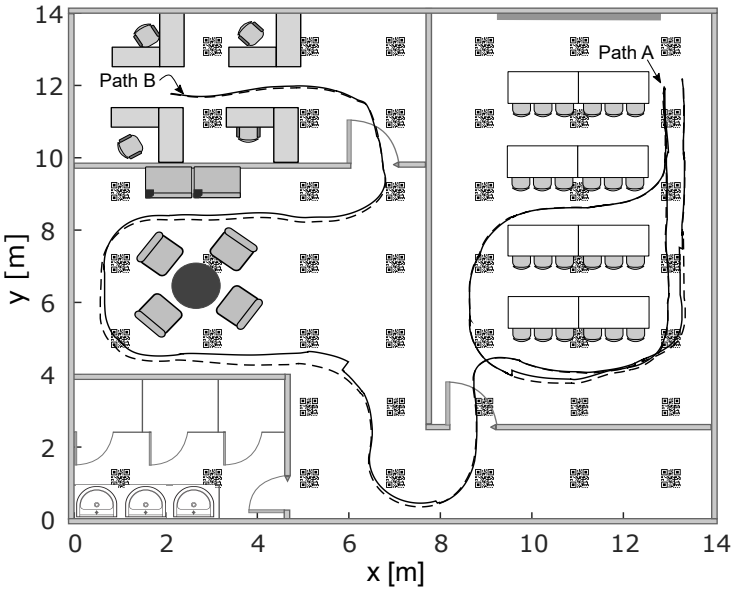


Figure 5.17: Sample paths (*Path A* and *Path B*) in a real-world scenario. Both the actual (dashed line) and the trajectory estimated by the EHF (solid line) are reported.

5.6 EHF in real environments

In order to evaluate the performances of the proposed EHF with the best weighting coefficients in a real environment, the EHF was implemented in the c-Walker and the robot was driven along two paths in the premises of the Department of Industrial Engineering of the University of Trento[1]. The map of a possible environment is shown in Figure 5.17. In this setup the distance between adjacent QR codes was set equal to $D = 2\ m$, as this choice provides a reasonable trade-off between performances and deployment time. *Path A* is a closed-loop trajectory starting and ending within the same room and in the same position. *Path B* instead starts in the upper left room and finishes in the room on the right side of the map. In Figure 5.17, both the estimated trajectory (solid line) and the ground truth (dashed line) are reported. The estimation errors of state variables x, y and θ in both cases are shown in Figure 5.18. This real experiment shows the obvious fact that the estimation error cannot converge asymptotically to zero due to the sporadic absolute position QR-based updates. For instance, in the central part of *Path B*, the length of the path actually traveled without QR code detection is much larger than the average value (i.e. in the order of some tens of meters). In such cases, the EHF is updated only by the gyroscopic platform for a long time. Hence, the whole state estimation uncertainty grows due to dead reckoning.

[1]These experiments were a part of the final Demo of DALi project.

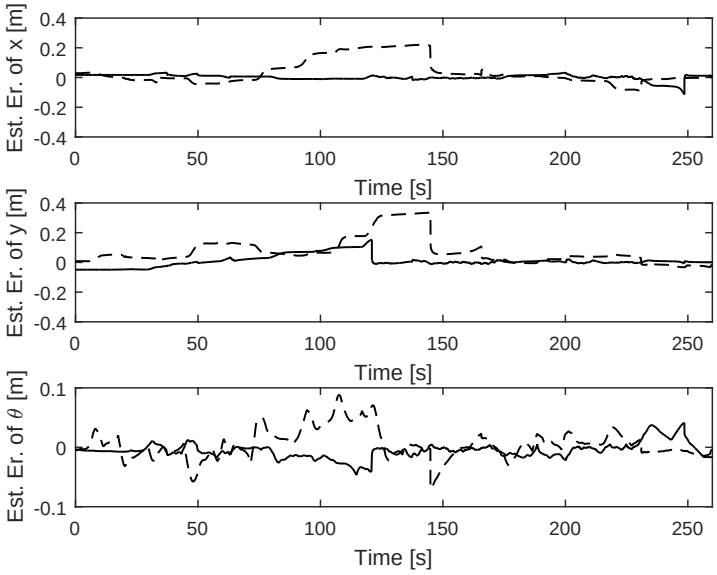


Figure 5.18: EHF estimation errors of state variables x, y and θ as a function of time when the *FriWalk* moves along *Path A* (solid line) and *Path B* (dashed line) depicted in Figure 5.17.

Chapter 6

Optimal Placement

In the previous chapters, the problem of indoor position tracking of wheeled robots using fusion of relative and absolute position measurements was addressed. When dealing with absolute position measurement systems, a common and general problem concerns with how to optimally place the landmarks in order to assure the detection of at least one landmark at every sample time of the sensor. This problem has been solved for omni-directional sensors, like RFID readers. However, to the best of our knowledge, no analytical solution exists in the literature for sensors with a limited angular detection range, such as the vision-based system used in the previous chapters.

6.1 An Introduction to Landmark Placement

Landmark placement is a prominent issue in every absolute position measurement technology. If the landmarks are located too far from each other, the localization system may occasionally miss them, thus reducing accuracy. On the other hand, if they are located too close, some of them would be redundant thus just increasing cost and complexity, especially in large places.

Depending on the application and on the chosen sensing technology, landmarks can be detected continuously or intermittently. In the former case, the optimal landmark placement is the one for which, ideally, just one landmark is detected at every sampling time. In the latter case, the number of landmarks can be much smaller, thus reducing deployment complexity and costs. However, a localization system should be also able to keep on tracking the agent's position with a reasonable level of confidence when no landmarks are detected. Of course, in both cases, the landmark placement problem strongly depends on the detection area of the sensor in use.

If the sensor detection area (SDA) is omni-directional, as shown in Figure 6.1, the agent's direction does not affect the landmark placement problem and only its location

should be analyzed. This problem has been addressed in several papers. In [51] a strip of RFID tags is deployed in such a way that, in every position, at least one RFID tag lies within the detection range of the on-board reader. The strip has an equilateral triangular shape, with the RFID tags being located in the vertices. In [104] authors show that,

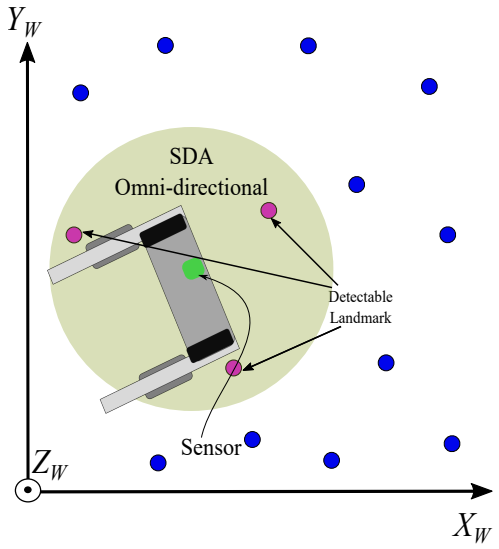


Figure 6.1: An agent with an omni-directional sensor moving in an environment with instrumented landmarks. The landmarks detection does not depend on the agent direction

in the case of landmark patterns consisting of equilateral triangles, the maximum side length is $\sqrt{3}$ times larger than the detection range. However, other kinds of patterns are also possible, and they can consist of up to 8-sided polygons [19]. A complete discussion about the landmark placement over rectangular patterns in the case of sensors with an omni-directional reading range is presented in [111]. An NP-hard placement problem was defined and analytically solved assuming the robot trajectory is known [9]. Such a solution was modified, using some observability constraints, to increase the robustness if some landmarks are missed [10].

Some solutions to the so-called "art gallery problem" can be also used for optimal landmarks placement [87, 31]. In the "art gallery problem" the area of a given environment (i.e. the art gallery) has to be partitioned into regions, in order to minimize the number of "guards" (i.e. the landmarks in this case) that can view every point of each region. The simulated annealing technique is frequently used to solve this kind of problems [31]. The art Gallery solution is mostly suitable for active landmarks. However, for systems in which the sensor is installed on the mobile agent, the art Gallery can just assure landmark detection over specific trajectories.

If the sensor reading range is not omni-directional, i.e. the sensor has limited detec-

tion angle, landmark detection depends not only on the relative position of agent and landmarks, but also on their relative orientation. This problem, clarified in Figure 6.2, happens, for instance, when well-known low-cost systems, such as simple cameras or ultrasonic sensors, are employed. However, a similar scenario holds even when wireless systems with strongly directional antennas are used. For this reason, the goal of this chapter is to find a general solution to the landmark placement problem, when the Sensor Detection Area (SDA) has a limited angular range and regardless to the agent's trajectory.

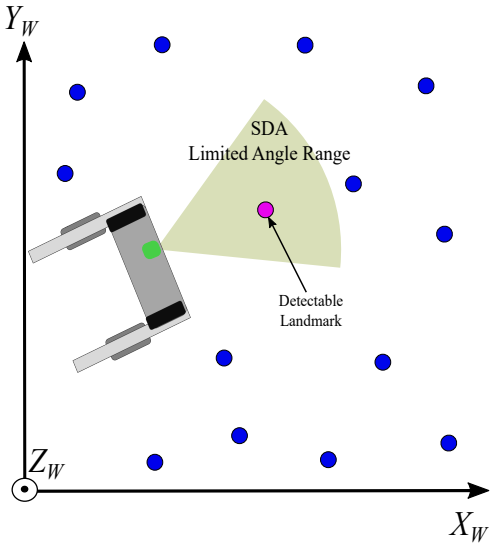


Figure 6.2: An agent with a limited angular range sensor moving in an environment with instrumented landmarks. The landmarks detection depends on the agent direction

Assuming to represent the SDA (e.g. the field of view of a standard RGB-D camera) with a polytope, first an optimal solution based on numerical techniques is introduced and discussed. Then, it will be shown that similar results can be achieved analytically.

6.2 Optimal Landmarks Placement

As explained previously, the goal is to determine the minimum number of landmarks to be deployed in a given environment so that, for any given configuration of the sensing system, at least one landmark lies within the SDA. This problem is often treated in the literature as a *tiling* problem, where the vertices of the tiles coincide with landmarks' positions. It is known that only three periodic, monohedral and regular tiling patterns exist in $\mathbb{R}^2$ (namely over the plane), i.e. triangles, squares and hexagons [5, 99]. Using triangle and square tiles (which are the easiest to deploy in practice) the vertices of all polytopes belong to an A^2 *point lattice* and a *square point lattice*, respectively. Thus, if

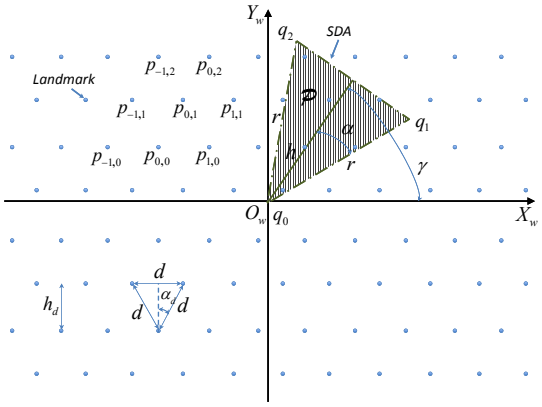


Figure 6.3: Points of an A^2 lattice representing the possible landmark positions in a wide-open, unbounded room. The sensor detection area (SDA) is represented by the shadowed region inside polytope $\mathcal{P}$.

the polytope $\mathcal{P}$ represents the SDA, the original optimization problem can be regarded as *"the problem of finding whether the polytope contains a lattice point"* for any possible position and orientation of $\mathcal{P}$ [99]. Limiting the analysis to the case of triangular tiles only (the extension to the square case is similar and is left for future work), a convenient way to represent the lattice points on a plane is to assume that one of the triangle sides is parallel to the X_w axis of the reference frame $\langle W \rangle = \{O_w, X_w, Y_w\}$, as shown in Figure 6.3. If $p_{0,0} = [x_0, y_0]^T$ denotes a given lattice reference point, the coordinates of any other point of the lattice can be expressed as

$$p_{i,j} = \begin{bmatrix} x_{i,j} \\ y_{i,j} \end{bmatrix} = p_{0,0} + \begin{bmatrix} j\frac{d}{2} + id \\ jh_d \end{bmatrix}, \quad \forall i, j \in \mathbb{Z}, \tag{6.1}$$

where d, $\alpha_d = \pi/6$ and $h_d = d\cos(\alpha_d)$ are the side length, the semi-angle and the height of any equilateral triangle, respectively. Observe that d is also the distance between every pair of adjacent landmarks. For the sake of simplicity, but without loss of generality, the polytope $\mathcal{P}$ defining the SDA is approximated with an isosceles triangle with a vertex angle of 2α, $\alpha \in (0, \pi/2)$ and height h (see Figure 6.3). In this case, the two equal sides of the triangular SDA have length $r = h/\cos(\alpha)$. Moreover, if γ denotes the orientation angle of the SDA with respect to axis X_w, the polytope is defined as the plane portion

$$\mathcal{P} \triangleq \left\{ \sum_{i=0}^{2} \lambda_i q_i \middle| \sum_{i=0}^{2} \lambda_i = 1, \lambda_i \in [0,1] \text{ for } i = 0,1,2 \right\}, \tag{6.2}$$

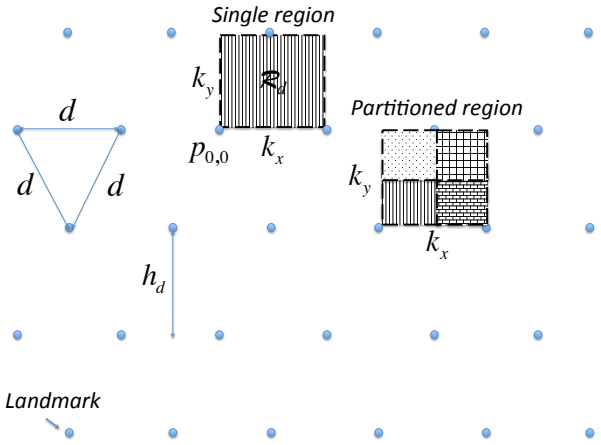


Figure 6.4: Possible landmark positions in the rectangle $\mathcal{R}_d$ and corresponding partitions.

where $q_0 = O_w = [0, 0]^T$,

$$q_1 = r \begin{bmatrix} \sin(\beta + \gamma) \\ -\cos(\beta + \gamma) \end{bmatrix}, \qquad q_2 = r \begin{bmatrix} \sin(\beta - \gamma) \\ \cos(\beta - \gamma) \end{bmatrix}, \tag{6.3}$$

are the SDA vertices and $\beta = \pi/2 - \alpha$ is the angle between segment $\overline{q_0 q_1}$ and axis X_w when $\gamma = \pi/2$, as shown in Figure 6.3.

Notice that, due to the symmetry and periodicity of the triangular lattice, in order to generate all the possible lattice positions, it is sufficient to move the reference point $p_{0,0}$ in the rectangle $\mathcal{R}_d$ with base d and height h_d, as pictorially explained in Figure 6.4. Therefore, with reference to the notation introduced above, the optimal landmarks placement problem can be formulated as follows, i.e.

Problem 1 *Maximize the landmark distance d such that $\forall p_{0,0} \in \mathcal{R}_d$ and $\forall \gamma \in [0, 2\pi)$ there exists at least one $p_{i,j} \in \mathcal{P}$ with $i, j \in \mathbb{Z}$.*

In the following subsections, first a numerical solution to solve this problem is presented. Then, a closed-form analytical solution is also derived.

6.2.1 Optimal Numerical Solution

According to Problem 1, a solution should be found $\forall p_{0,0} \in \mathcal{R}_d$. Hence, we can rewrite

$$p_{0,0} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} k_x d \\ k_y h_d \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} k_x d \\ k_y d c_6 \end{bmatrix},$$

where $c_6 = \cos(\pi/6)$ and $k_x, k_y \in [0, 1]$. This parametrization is shadowed as *single region* in Figure 6.4. Given that $[x_0, y_0]^T$ is generic, we can choose $[x_0, y_0]^T = [0, 0]^T$ for

simplicity. As a consequence, it follows from (6.1) that

$$p_{i,j} = \begin{bmatrix} k_x d + j\frac{d}{2} + id \\ k_y dc_6 + jdc_6 \end{bmatrix} = d \begin{bmatrix} \frac{1}{2}(2k_x + j + 2i) \\ c_6(k_y + j) \end{bmatrix}. \tag{6.4}$$

According to the definition of polytope (6.2), a point $p_{i,j} \in \mathcal{P}$ if and only if $\exists \lambda_1, \lambda_2 \in [0,1]$ such that $\lambda_1 q_1 + \lambda_2 q_2 = p_{i,j}$, with $\lambda_1 + \lambda_2 \leq 1$. Recalling (6.3), if we define

$$Q_\gamma = [q_1, q_2] = r \begin{bmatrix} \sin(\beta+\gamma) & \sin(\beta-\gamma) \\ -\cos(\beta+\gamma) & \cos(\beta-\gamma) \end{bmatrix}, \tag{6.5}$$

and $\Lambda = [\lambda_1, \lambda_2]^T$, it follows that

$$Q_\gamma \Lambda = p_{i,j} \Rightarrow \Lambda = Q_\gamma^{-1} p_{i,j}, \tag{6.6}$$

Since $\det(Q_\gamma) = r\sin(2\beta)$, the inverse of Q_γ always exists and can be determined as

$$Q_\gamma^{-1} = \frac{1}{r\sin(2\beta)} \begin{bmatrix} \cos(\beta-\gamma) & -\sin(\beta-\gamma) \\ \cos(\beta+\gamma) & \sin(\beta+\gamma) \end{bmatrix}, \tag{6.7}$$

the constraints on coefficients λ_1 and λ_2 can be expressed as

$$\begin{aligned} \lambda_1 \geq 0 \quad &\Leftrightarrow \begin{bmatrix} 1 & 0 \end{bmatrix} \Lambda \geq 0 \Leftrightarrow \begin{bmatrix} 1 & 0 \end{bmatrix} Q_\gamma^{-1} p_{i,j} \geq 0, \\ \lambda_2 \geq 0 \quad &\Leftrightarrow \begin{bmatrix} 0 & 1 \end{bmatrix} \Lambda \geq 0 \Leftrightarrow \begin{bmatrix} 0 & 1 \end{bmatrix} Q_\gamma^{-1} p_{i,j} \geq 0, \\ \lambda_1 + \lambda_2 \leq 1 \quad &\Leftrightarrow \begin{bmatrix} 1 & 1 \end{bmatrix} \Lambda \leq 1 \Leftrightarrow \begin{bmatrix} 1 & 1 \end{bmatrix} Q_\gamma^{-1} p_{i,j} \leq 1, \end{aligned} \tag{6.8}$$

which yields to

$$\begin{cases} \cos(\beta-\gamma)x_{i,j} - \sin(\beta-\gamma)y_{i,j} \geq 0, \\ \cos(\beta+\gamma)x_{i,j} + \sin(\beta+\gamma)y_{i,j} \geq 0, \\ (\cos(\beta-\gamma) + \cos(\beta+\gamma))x_{i,j} + (\sin(\beta+\gamma) - \sin(\beta-\gamma))y_{i,j} \leq r\sin(2\beta). \end{cases} \tag{6.9}$$

Hence, after some algebraic steps it follows that

$$\begin{cases} \cos(\beta-\gamma)x_{i,j} - \sin(\beta-\gamma)y_{i,j} \geq 0, \\ \cos(\beta+\gamma)x_{i,j} + \sin(\beta+\gamma)y_{i,j} \geq 0, \\ \cos(\gamma)x_{i,j} + \sin(\gamma)y_{i,j} \leq \dfrac{r}{2}\dfrac{\sin(2\beta)}{\cos(\beta)}. \end{cases} \tag{6.10}$$

Thus, by replacing the elements $x_{i,j}$ and $y_{i,j}$ of (6.4) as well as $\frac{r}{2}\frac{\sin(2\beta)}{\cos(\beta)} = r\sin(\beta) = h$ into (6.10), the following system of inequalities results

$$\begin{cases} \dfrac{\cos(\beta-\gamma)}{2}(2k_x+j+2i)-\sin(\beta-\gamma)c_6(k_y+j)\geq 0, \\ \dfrac{\cos(\beta+\gamma)}{2}(2k_x+j+2i)+\sin(\beta+\gamma)c_6(k_y+j)\geq 0, \\ d\left[\dfrac{\cos(\gamma)}{2}(2k_x+j+2i)+\sin(\gamma)c_6(k_y+j)\right]\leq h. \end{cases} \tag{6.11}$$

Therefore, a point $p_{i,j} \in \mathcal{P}$ if and only if $\exists i, j \in \mathbb{Z}$ satisfying (6.11) and to solve Problem 1, we have to find a solution to system (6.11) for any possible value of variables γ, k_x and k_y. Notice that:

1. All the inequalities of (6.11) are linear in $i, j \in \mathbb{Z}$ for given values of γ, k_x and k_y.

2. Since the coefficients of (6.11) depend on periodic trigonometric functions, just the values of γ in the interval $[0, \pi/2]$ should be taken into consideration. This is a direct consequence of the regular, periodic lattice structure. While the possible values of γ are infinite, being the range of variation limited and $i, j \in \mathbb{Z}$, in practice just a finite amount of γ values (e.g. chosen with a resolution of $\pi/40$) can be explored to find the optimal solution.

3. Given that the objective of Problem 1 is to maximize d, the first two inequalities in (6.11) provide the constraints to the possible values of $i, j \in \mathbb{Z}$, whereas the third inequality represents the actual cost function to optimize. By adding the first two inequalities and by using basic trigonometric functions properties, it can be easily proved that $\frac{\cos(\gamma)}{2}(2k_x + j + 2i) + \sin(\gamma)c_6(k_y + j) > 0$. Therefore, since $\cos(\gamma) \geq 0$, $\sin(\gamma) \geq 0$ and $k_x, k_y \in [0, 1]$, it ensues immediately that the maximum of d for a given γ (denoted as d_γ in the following) is obtained for those value of $i, j \in \mathbb{Z}$ minimizing the cost function $f_\gamma J$, where

$$f_\gamma = \left[\cos(\gamma) \quad \tfrac{\cos(\gamma)}{2} + \sin(\gamma)c_6\right], \tag{6.12}$$

and $J = [i, j]^T$.

In light of the previous remarks, we can rewrite in a compact, matrix form the first two linear inequalities of (6.11) as $A_1 J \geq B_1 K_1$ and $A_2 J \geq B_2 K_2$, respectively, where

$$A_1 = \left[\cos(\beta-\gamma) \quad \tfrac{\cos(\beta-\gamma)}{2} - \sin(\beta-\gamma)c_6\right]$$

$$A_2 = \left[\cos(\beta+\gamma) \quad \tfrac{\cos(\beta+\gamma)}{2} + \sin(\beta+\gamma)c_6\right],$$

$$B_1 = \begin{bmatrix} -\cos(\beta-\gamma) & \sin(\beta-\gamma)c_6 \end{bmatrix},$$
$$B_2 = \begin{bmatrix} -\cos(\beta+\gamma) & -\sin(\beta+\gamma)c_6 \end{bmatrix},$$

and $K_1 = [k_{x_1}, k_{y_1}]^T$ and $K_2 = [k_{x_2}, k_{y_2}]^T$ are the values that, in the worst case, maximize B_1K and B_2K, respectively, $K = [k_x, k_y]^T$ being the vector of generic coefficients. Therefore, for a given $\gamma \in [0, \pi/2]$, the solution $i^\star, j^\star \in \mathbb{Z}$ of the following optimal problem

$$\begin{aligned} \min_{i,j} \ & f_\gamma J \\ \text{s.t.} \ & A_1 J \geq B_1 K_1, \\ & A_2 J \geq B_2 K_2, \\ & i, j \in \mathbb{Z}, \end{aligned} \tag{6.13}$$

can be used to compute the maximum distance d_γ for a given γ, i.e.

$$d_\gamma = \frac{2h}{\cos(\gamma)(2k_x + j^\star + 2i^\star) + 2\sin(\gamma)c_6(k_y + j^\star)}. \tag{6.14}$$

It is worthwhile to note that a solution to Problem (6.13) always exists since $\det([A_1^T, A_2^T]) = \sin(2\beta)c_6 \neq 0$. However, the pair $i^\star, j^\star \in \mathbb{Z}$ defines a point lattice $p_{i^\star,j^\star} \in \mathcal{P}$, $\forall k_x, k_y \in [0, 1]$. In other words, the solution determines a single point that belongs to the SDA $\forall p_{0,0} \in \mathcal{R}_d$. This is clearly an overkill, since it is sufficient that at least *one* point belongs to the SDA, even if this is not *the same* point $\forall k_x, k_y \in [0, 1]$. To address this issue, $\mathcal{R}_d$ can be partitioned into smaller sub-regions (i.e. by bisecting k_x and k_y iteratively) in order to compute the optimal pair $i^\star, j^\star \in \mathbb{Z}$ for each sub-region (see the *Partitioned region* in Figure 6.4 for reference). In this way, if $\mathcal{J}^\star$ denotes the set of optimal $i^\star$, $j^\star$ pairs of all sub-regions, the optimal distance between landmarks for a given γ results

$$d_\gamma^\star = \min_{i^\star, j^\star \in \mathcal{J}^\star} \frac{2h}{\cos(\gamma)(2k_x + j^\star + 2i^\star) + 2\sin(\gamma)c_6(k_y + j^\star)}. \tag{6.15}$$

Finally, the optimal solution to Problem 1 is given by

$$d^\star = \min_{\gamma \in [0, \pi/2]} d_\gamma^\star. \tag{6.16}$$

The results of some simulations confirming the validity of the proposed optimal solution are reported in Section 6.3.

6.2.2 Optimal Analytical Solution

This Section reports an analytical expression of the optimal distance between landmarks, when an A^2 lattice in a wide-open unbounded room is considered. Let $b = 2r\sin(\alpha)$ be

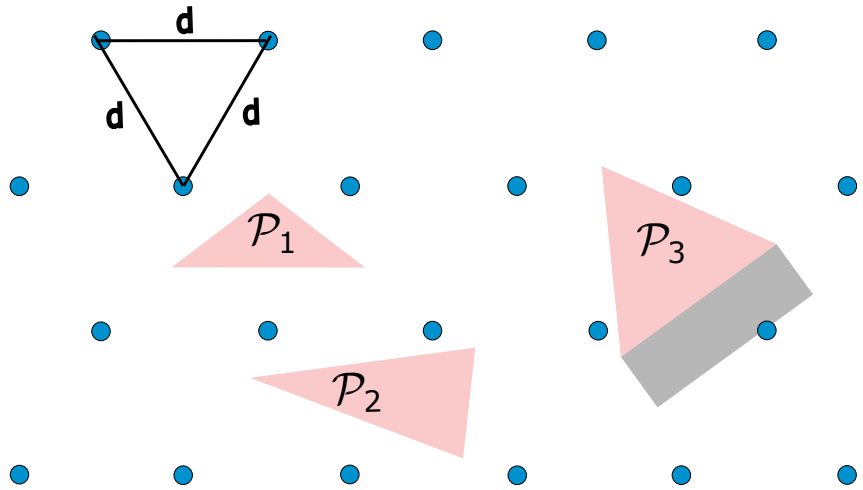


Figure 6.5: Pictorial examples of missed landmark detection, when the geometrical constraints $d \leq r$ (e.g. $\mathcal{P}_1$) and $d \leq b$ (e.g. $\mathcal{P}_2$) are not satisfied.

the SDA maximum width, i.e. the length of the base of the isosceles triangle $\mathcal{P}$. In order to solve Problem 1, two geometrical constraints must be satisfied, i.e. $d \leq r$ and $d \leq b$. Indeed, if these constraints are not met, at least one triple of values k_x, k_y and γ exists such that the sensor cannot detect any landmark (e.g. $\mathcal{P}_1$ and $\mathcal{P}_2$ in Figure 6.5). This can happen in some situations even if these constraints are met, like $\mathcal{P}_3$ in Figure 6.5. However, in the case of $\mathcal{P}_3$ since the SDA base is larger than d ($b > d$), the base is obviously situated inside more than one tile and hence there exists a rectangular area in front of the base comprising one vertex (landmark).

Let us consider a *virtual* sensor with a triangular SDA included into $\mathcal{P}$. Similarly to (6.2), the SDA of the virtual sensor is defined as follows, i.e.

$$\mathcal{P}^v \triangleq \left\{ \sum_{i=0}^{2} \lambda_i q_i^v \Big| \sum_{i=0}^{2} \lambda_i = 1, \lambda_i \in [0,1] \text{ for } i = 0,1,2 \right\}, \tag{6.17}$$

where $q_0^v = q_0$ and

$$q_1^v = r^v \begin{bmatrix} \sin(\beta+\gamma) \\ -\cos(\beta+\gamma) \end{bmatrix}, q_2^v = r^v \begin{bmatrix} \sin(\beta-\gamma) \\ \cos(\beta-\gamma) \end{bmatrix},$$

with $r^v \leq r$, $d \leq r^v$ and $d \leq b^v = 2r^v \sin(\alpha)$, in accordance with the constraints specified above. Let $\mathcal{R}^v$ be the rectangle, with base b^v and height $H = h - h^v = (r - r^v)\cos(\alpha)$, which lies just beyond the SDA of the virtual sensor (see Figure 6.6 for reference). Observe that, in general, $\{\mathcal{P}^v \cup \mathcal{R}^v\} \subseteq \mathcal{P}$. So $\{\mathcal{P}^v \cup \mathcal{R}^v\}$ can be regarded as an inner approximation of polytope $\mathcal{P}$, which becomes increasingly accurate as $r^v \to r$. In light of this approximation, a sub-optimal version of Problem 1 can be formulated as follows, i.e.

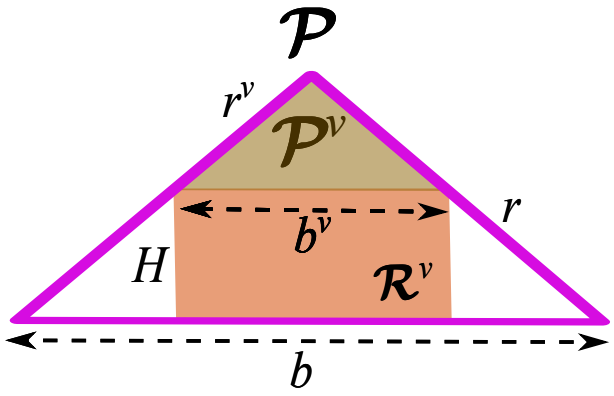


Figure 6.6: The rectangle $\mathcal{R}^v$ in front of the FoV $\mathcal{P}^v$.

Problem 2 *Maximize the landmark distance d such that $\forall p_{0,0} \in \mathcal{R}_d$ and $\forall \gamma \in [0, 2\pi)$ there exists at least one pair $i, j \in \mathbb{Z}$ with $p_{i,j} \in \{\mathcal{P}^v \cup \mathcal{R}^v\}$.*

To find an analytical solution to this problem, first of all notice that, since $b_v \leq b$, then $\max d = \min(r, b^v)$. Given that α, b and r are known parameters of the sensor, but b^v is unknown, using simple geometric arguments, it can be shown that

$$b^v = 2\tan(\alpha)(r\cos(\alpha) - H). \tag{6.18}$$

Therefore, in order to maximize b^v it is sufficient to minimize H. Let us suppose that, for a given choice of k_x, k_y and γ, then $p_{i,j} \notin \mathcal{P}^v$ for any $i, j \in \mathbb{Z}$ (otherwise Problem 2 would be straightforwardly solved). Under this assumption, we need to have one landmark in $\mathcal{R}^v$. This in turn implies that $\mathcal{R}_d \subseteq \mathcal{R}^v$, where the areas of $\mathcal{R}_d$ and $\mathcal{R}^v$ are equal to $d \cdot h_d$ and $H \cdot b^v$, respectively. Since $d \leq b^v$, the minimum value of H ensuring that $\mathcal{R}_d \subseteq \mathcal{R}^v$ is

$$H = h_d = d\frac{\sqrt{3}}{2}. \tag{6.19}$$

Thus, by plugging (6.19) into (6.18), it finally results that

$$d \leq d^\dagger = b^v = r\frac{2\sin(\alpha)}{1 + \sqrt{3}\tan(\alpha)}. \tag{6.20}$$

where $d^\dagger$ denotes the analytical solution to Problem 2. Notice that $0 < d^\dagger < r$, because $\alpha \in (0, \pi/2)$.

6.3 Simulation-based Validation of Numerical and Analytical Optimal Solutions

In order to confirm that the optimal landmark distances obtained both numerically and analytically are correct and converge to the same solution, some meaningful simulation

have been performed. Figure 6.7 shows the normalized optimal landmark distances (i.e. $\frac{d^\star}{r}$ and $\frac{d^\dagger}{r}$) as a function of the sensor angular semi-range $\alpha \in (0, \pi/2)$. In the numerical case, three sets of results are reported for different partitions of $\mathcal{R}_d$ (i.e. assuming to find a solution in the entire $\mathcal{R}_d$, in 8 sub-regions and in 64 sub-regions of $\mathcal{R}_d$, respectively). Notice that as the number of partitions used to compute $d^\star$ grows, the sub-optimal nu-

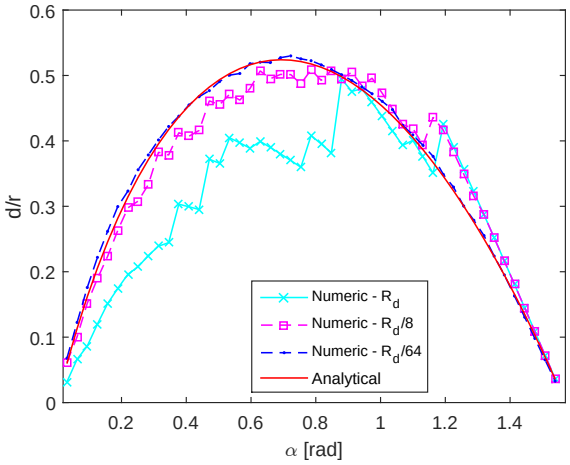


Figure 6.7: Optimal landmark distances (normalized by r) computed numerically and analytically as a function of the sensor angular semi-range α. The numerical results refer to three different partitions of $\mathcal{R}_d$.

merical values exhibit smaller fluctuations (due to the finer granularity of the regions explored) and ultimately they converge to the analytical results, as expected. Observe also that, with a partition of $\mathcal{R}_d$ into 64 sub-regions, analytical and numerical results are no longer distinguishable.

In addition, Figure 6.8 reports dual results when the optimal landmark distance as a function of $\alpha \in (0, \pi/2)$ is normalized by h (i.e. $\frac{d^\star}{h}$ and $\frac{d^\dagger}{h}$). Notice that the trend of the curves in Figs. 6.7 and 6.8 is completely different, although numerical and analytical results are consistent. This is due to an essential geometric difference in the two cases. In Figure 6.7 the SDA triangle is indeed inscribed within a circle of constant radius r. This implies that as α changes the actual sensor range h is not fixed, but it reaches a maximum for $\frac{2\pi}{9}$ and then it decreases as α approaches $\frac{\pi}{2}$. On the contrary, in Figure 6.8 the SDA triangle has a constant height h, and the value of the circle radius r steadily increases with α. As a result, the normalized optimal landmark distances grow monotonically, as well. Consider that both situations may occur in real scenarios, as they depend on the setup of the sensing system (e.g. camera orientation with respect to the floor). Observe also that, in both cases, when $\mathcal{R}_d$ is partitioned into 64 sub-regions, analytical and numerical

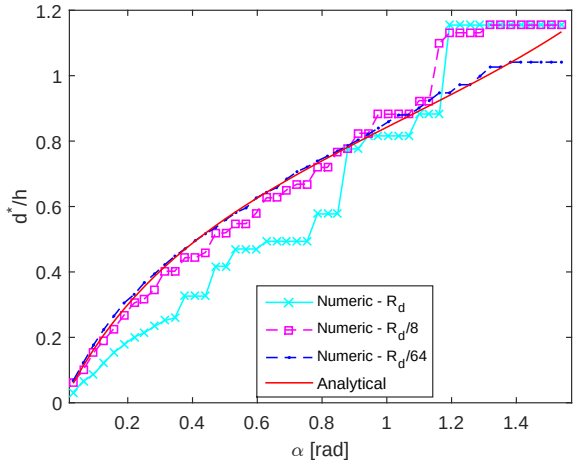


Figure 6.8: Optimal landmark distances (normalized by h) computed numerically and analytically as a function of the sensor angular semi-range α. The numerical results refer to three different partitions of $\mathcal{R}_d$.

Table 6.1: Optimal landmark distances (computed analytically and normalized by r) and minimum values of factor δ for which no landmark is detected in at least one out of 10^5 randomly generated positions and orientations of the sensor.

$\boldsymbol{\alpha}$ [rad]	0.2	0.4	0.6	0.8	1	1.2	1.4
$\boldsymbol{\frac{d^\dagger}{r}}$	0.32	0.46	0.53	0.53	0.46	0.35	0.18
$\boldsymbol{\delta}$	1.08	1.02	1.02	1.02	1.02	1.01	1.01

results are hardly distinguishable.

In order to evaluate more clearly to what extent the estimated landmark distance values are close to the optimal ones, some Monte Carlo simulations (of 10^5 runs each for given values of α) have been performed by randomly changing sensors' position and orientation over different lattices of type A^2, in which the distance between landmarks was set purposely larger than $d^\dagger$ by a variable factor δ. Tab. 6.1 reports both the optimal values of $\frac{d^\dagger}{r}$ and the minimum values of factor δ for which if we set $d = \delta \cdot d^\dagger$, there exists at least one configuration of the sensor in which no landmark is detected. It shows that for very small values of α, $d^\dagger$ is slightly far from the optimal d, e.g. approximately 8% for $\alpha = 0.2$ rad, but it tends to the optimal value when α increases.

6.4 Simulation Results for the c-Walker

In order to evaluate the impact of optimal landmark placement on localization accuracy in a real case study, the results of some Monte Carlo simulations in two different indoor environments are reported in the following, i.e. a large wide-open room without any obstacle, and a more realistic scenario based on the map of the Department of Information Engineering and Computer Science (DISI) of the University of Trento. The main parameters of the camera's SDA are supposed to be: $r \approx 4$ m and $\alpha \approx \pi/6$ rad. Thus, it follows from (6.20) that $d^\dagger \approx 2$ m.

6.4.1 Wide-Open Room

In the ideal case of a wide-open room without obstacles where an A^2 lattice of landmarks is deployed on the floor with distances between adjacent landmarks given by (6.16) or (6.20), the front camera of the *c-Walker* is always able to detect at least one landmark, regardless of camera's position and orientation. To verify this, 200 random-walk trajectories of 180 s each have been generated within a 10×10 m wide-open room. The results obtained in this case are comparable with those reported in Tab. 6.1. Indeed, by increasing the distance between adjacent landmarks by just a few percent, it may happen that no landmarks are detected. To evaluate the landmark detection, $d^\dagger$ was multiplied by a coefficient δ, i.e. $d = \delta d^\dagger$, and the probability of not detecting any landmark is computed by finding the samples in which no landmark is detected during all 100 trajectories. Figure 6.9 shows the probability for $\delta \in [1, 1.5]$.

It is worthwhile to note that setting $d = d^\dagger$ ensures that *at least one* landmark is in view for any position and orientation of the robot. Therefore, in general, more than one landmark can be actually detected. Figure 6.10 reports the average number of detected landmarks versus δ, with $\delta \cdot d^\dagger$ being the parametric distance between adjacent landmarks on the lattice considered. It is evident that more than 2 landmarks can be detected on average when $d = d^\dagger$.

Figure 6.11 shows the root mean square (RMS) estimation errors associated with variables (x, y, θ) as a function of time, when 200 simulated trajectories are considered. Observe that all RMS patterns quickly converge to the respective lower bounds that can be obtained using the EKF described in Chapter 4. Notice that these lower bounds can be reached if and only if the optimal landmark deployment is adopted, i.e. if there is at least one visual landmark inside the SDA of the camera.

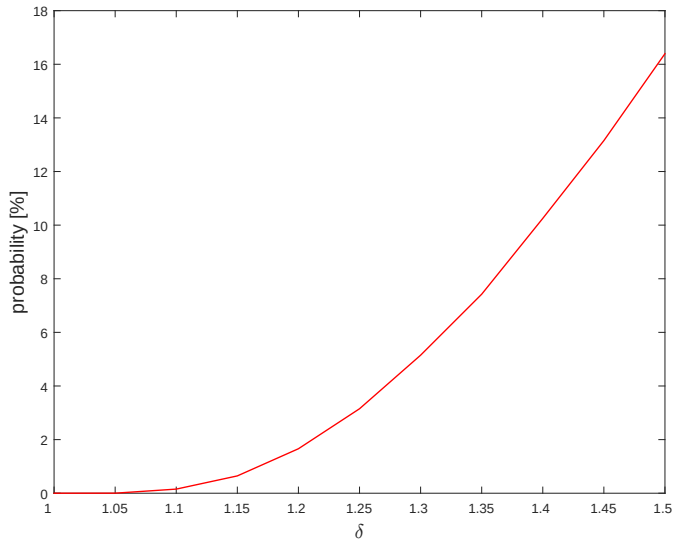


Figure 6.9: Probability of being in a position where no landmark can be detected for various adjacent landmark distances $\delta \cdot d^{\dagger}$ in the case of wide-open room with no obstacles

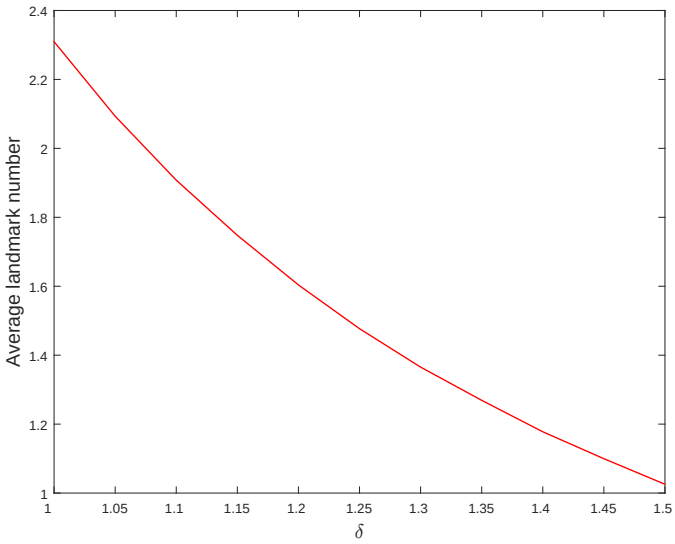


Figure 6.10: Average number of detected landmarks for various adjacent landmark distances $\delta \cdot d^{\dagger}$ in the case of wide-open room with no obstacles.

6.4.2 Realistic Environment

To evaluate the method in a real environment, three users are supposed to move simultaneously in the building of DISI[1]. Department. For each user, 50 trajectories, whose initial

[1]Department of Information Engineering and Computer Science, University of Trento

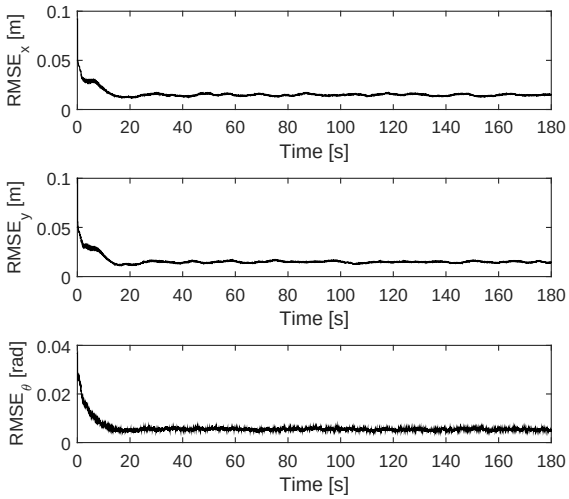


Figure 6.11: RMS estimation error of robot's planar Cartesian coordinates and orientation computed over 200 random trajectories in the case of a wide-open room with no obstacles.

and final destinations are generated randomly, have been computed using a Social Force Model described in [44] to simulate human-like trajectories and avoid collisions.

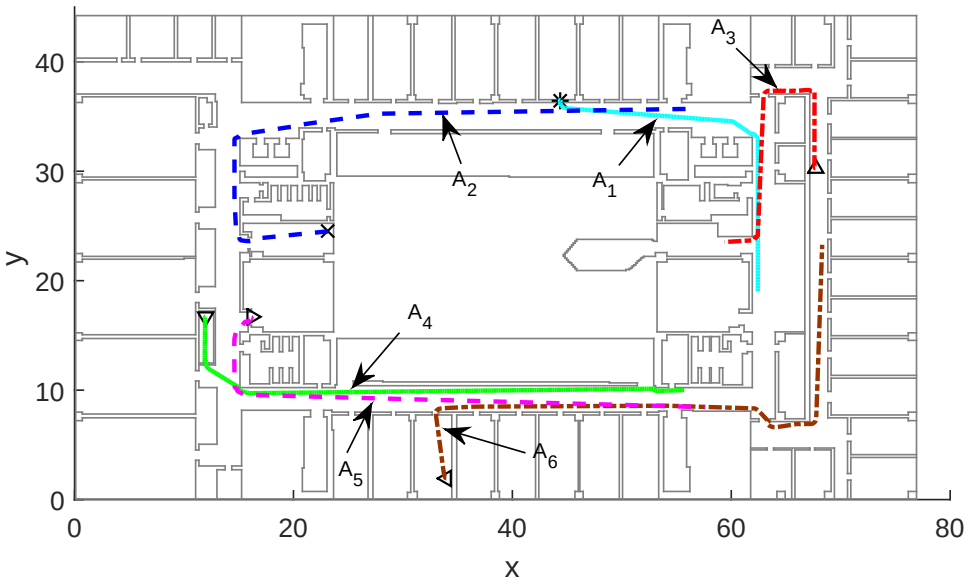
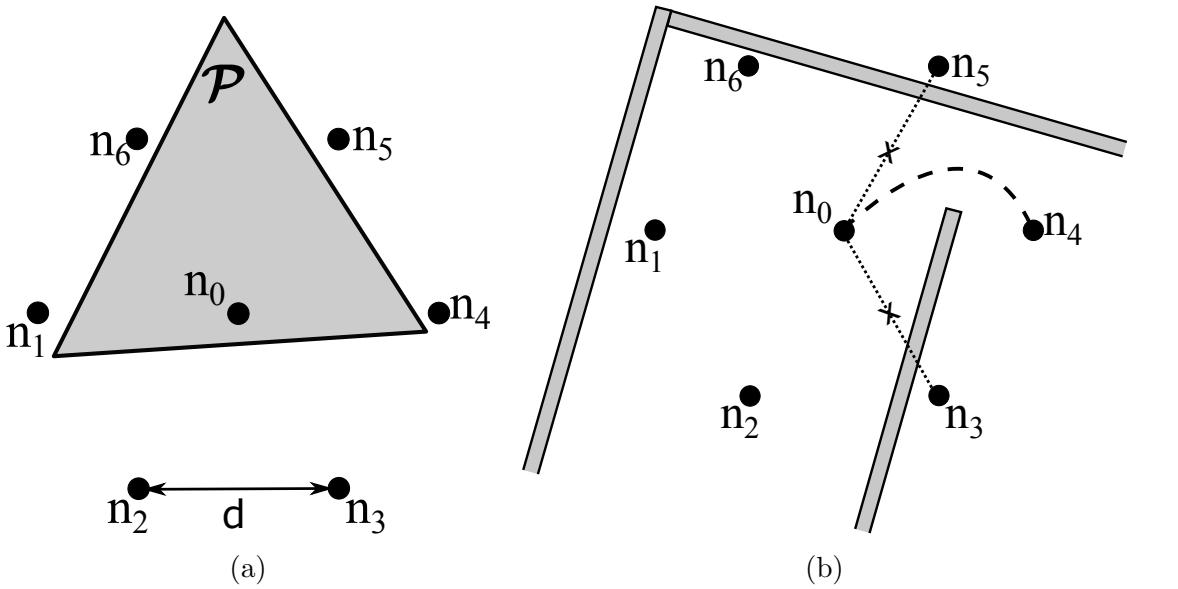


Figure 6.12: Trajectories planned for six agents' moving in the premises of the Department of Information Engineering and Computer Science of the University of Trento.

Figure 6.12 shows the map of DISI along with one example of six users' trajectories.



(a) (b)

Figure 6.13: Qualitative relationship between the SDA of the robot's camera, the position of a detected landmark and its six neighbors over a portion of A^2 lattice in the case of wide-open room (a) and obstacles (e.g. walls) (b).

While initial position and final destinations are generated randomly, the main difference with respect to the wide-open room case is that now the optimal landmarks placement is affected by the presence of obstacles and walls. Anytime a point of the lattice is located inside a wall or an obstacle reported in the DISI map, obviously the corresponding landmark is not available in practice. This problem can be partially addressed by shifting the reference landmark position $p_{0,0} \in \mathcal{R}_d$ until the maximum number of available landmarks is reached. However, while in the wide-open room case, anytime the camera detects a landmark, any one of its six neighbors can be detected next (as depicted in Figure 6.13(a)), the presence of walls and obstacles can make the transition from one landmark to another impossible or much longer (e.g. because the trajectory is constrained by a wall as shown in Figure 6.13(b)).

This fact influences RMS estimation errors even if the layout of landmarks is as close as possible to the optimal one. To evaluate the impact of obstacles on localization accuracy, the RMS estimation errors over time associated with variables (x, y, θ) and averaged over 200 trajectories have been computed for three different values of r (i.e. 1 m, 4 m and 8 m) and are reported in Tab. 6.2 in the case of wide-open room and DISI premises, respectively.

It is clear that, in the case of a wide-open room, the SDA size does not affect localization accuracy significantly, as lattice optimality for a given SDA is preserved. On the contrary, when walls and obstacles are present, the RMS position errors tend to increase

Table 6.2: Average RMS estimation errors associated with variables (x, y, θ) for different values of r and in the case of wide-open room and DISI premises, respectively.

	Wide-open room			DISI premises		
r [m]	1	4	8	1	4	8
$RMSE_x$ [cm]	1.5	1.6	1.7	2.2	3.3	4.2
$RMSE_y$ [cm]	1.5	1.6	1.7	1.5	2.2	2.9
$RMSE_\theta$ [mrad]	8.0	8.0	8.0	24.0	25.0	25.0

as the camera SDA grows. This is indeed due to the fact that the probability of detecting a landmark decreases from 96% when $r = 1$ m to 87% when $r = 8$ m. To emulate and to analyze the effect of the presence obstacles more in depth, an increasing amount of landmarks has been randomly removed from the optimal layout, recomputing the RMS position errors every time. Figure 6.14 shows the RMS position error accumulated, on average, along the path between pairs of subsequent landmarks, when an increasing percentage of landmarks cannot be detected. Notice that the RMS position error increases as expected, in accordance with the results reported in Tab. 6.2.

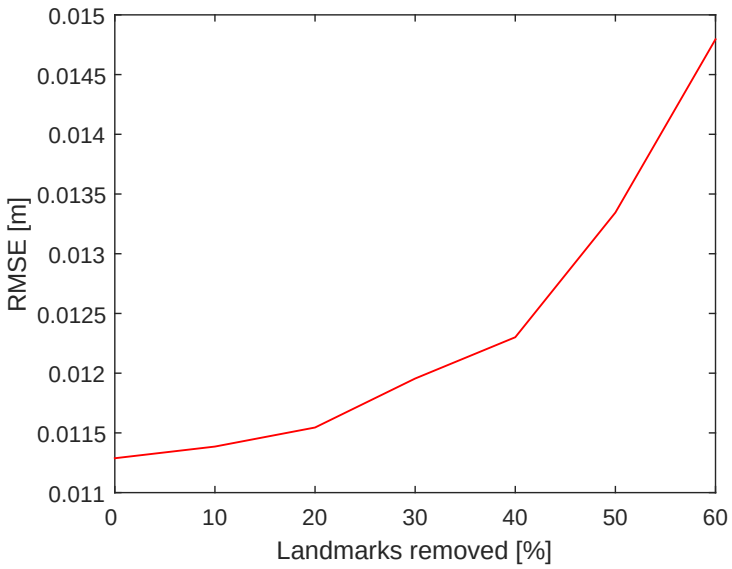


Figure 6.14: RMS position errors between two subsequent landmark detections as a function of the percentage of landmarks randomly removed from the optimal layout.

Chapter 7

Collaborative Localization

In the previous chapters, the localization problem was defined and solved for individual wheeled robots. However, if several agents are moving in the same environment, they can be another source of information for each other. In other words, they can inform each other about what happens in their surroundings. In this case, if one agent can estimate its own relative position with respect to the others and it receives the others location, it can reconstruct its own position as well. This collaboration between different agents can be combined with the self-localization system, described in previous chapters, to achieve higher estimation accuracy.

The localization of multiple mobile agents is typically a difficult problem if it is addressed in a centralized way, especially in non-line-of-sight (NLOS) conditions and when multiple targets are present in the same environment. However, when each agent has its own autonomous localization system, the problem can be tackled in a distributed way. In this situation, the presence of multiple *agents* can possibly turn into an advantage, provided that different robots are able to cooperate. This general idea, often called *synergic* or *collaborative* localization, has proved to be successful in different contexts involving groups of robots [97, 49].

An early study on collaborative localization is presented in [85]. In [34] the authors envision a fully wireless collaborative localization system based on the potential ability of clusters of 4G mobile devices to measure their reciprocal distances through a hybrid time of arrival/angle of arrival (TOA/AOA) technique. The case of collaborative localization of wireless mobile platforms has been also addressed in [15], where the so-called *parallel projection method* is used to improve the localization accuracy in NLOS conditions. Taniuchi et al. suggest using a spring model to reduce the pose estimation uncertainty associated with distance measures obtained using WiFi and Bluetooth RSS data [98].

In the field of robotics, Fox et al. propose a Markov-based probabilistic method in which each robot's belief about its own position is refined as soon as other robots are

detected [33]. A different Markovian approach is adopted in [4]. In this case, first the egocentric measurement data are fused locally to create a Markov chain of robot pose estimates. Then, both inter-robot measurement data and state estimates are transferred to a central server, where localization is refined by minimizing the mean square error of agents' positions.

An alternative statistical method for collaborative localization is instead described in [82]. This relies on a decentralized, real-time particle filter coupled with a reciprocal sampling algorithm to reduce the overall computational burden. In spite of the accuracy improvements achieved by applying computationally demanding optimization strategies to the collaborative localization problem [52], the simplest general technique for fully-distributed, multi-robot localization is still the extended Kalman filter (EKF) [85]. In [3] the update step of a distributed EKF is modified by an algorithm preventing data reuse in order to avoid inconsistent (i.e. overconfident) covariance estimates.

Panzieri et al. address the collaborative localization problem by means of an interlaced extended Kalman filter (IEKF) [72, 73] based on the fact that the state evolution of each robot and its observation depend also on the state of other robots. The IEKF is inherently distributed, computationally acceptable and easy to implement. Therefore, [72, 73] was the starting point for the research work proposed in this chapter. However, the implementation of the original IEKF disregards the possibility of Event-based observations while in real environments it is unlikely to detect other robots at every time. Hence, in this chapter the IEKF is modified by separating the individual and collaborative localization with the latter being used only when proper information from other agents is available. In other words, the proposed solution is a two-layer localization approach in which layer 1 relies on the local sensing features of each robot to estimate its own pose and layer 2 instead exploits the results of layer 1 in order to refine the local estimates through an IEKF. An overview of the proposed two-layer solution is depicted in Figure 7.1.

A key challenge in collaborative localization of mobile agents is the sensing technology used to observe the relative position of nearby agents. This chapter reports several simulation results comparing two different techniques to measure the relative position of pairs of agents *a*) a low-rate RGB-D camera and *b*) a wireless ranging system. The former provides more accurate measurements but at a very low rate, while the latter is less accurate but returns data at a higher rate and covers a larger area.

7.1 System Model Description

Collaborative localization of a team of agents refers to the ability to refine the position and the heading estimated by each agent in a common reference frame by using both local

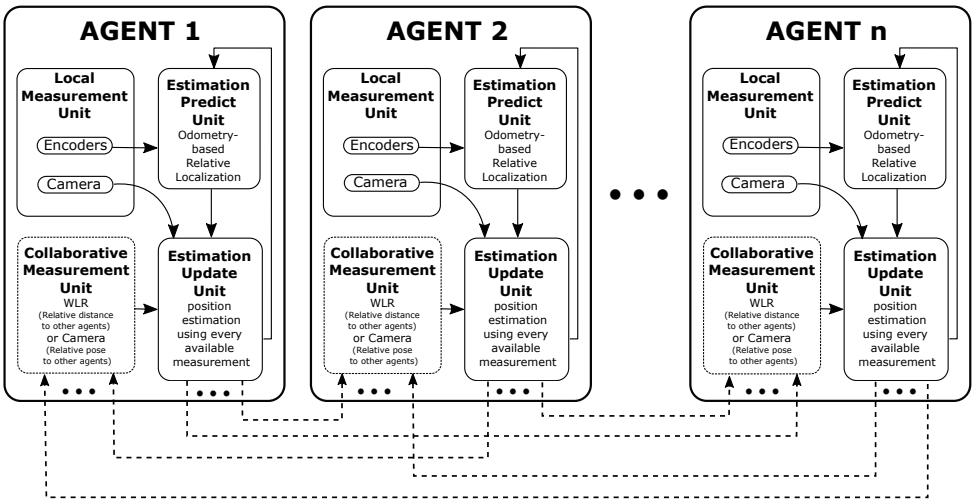


Figure 7.1: An overview of the proposed two-layer collaborative localization system. The self-localization process can be run independently if no collaborative data is available

positioning data and relative distance and/or attitude measures between pairs of devices. The main underlying assumptions are summarized below.

1. N agents can move freely in a large room. The dynamic of each agent does not depend on any other agent, since each user may act independently. The only constraint to robot motion is collision avoidance.

2. The state of each agent i (with $i = 1, \ldots, N$) at time kT_s is represented by vector $\mathbf{p}_k^{(i)} = [x_k^{(i)}, y_k^{(i)}, \theta_k^{(i)}]^T$

3. Each agent is able to estimate its own state autonomously (namely without the help of other agents) using the EKF technique described in Chapter 4.

4. Besides the sensors used by each robot for its own local state estimation, every agent is supposed to be equipped with two alternative types of exteroceptive sensors for collaborative localization, i.e. either an omni-directional wireless ranging system (*case A*) or a front RGB-D camera (e.g. a Kinect) (*case B*).

   In *case A*, the ranging system is used to measure just the distance between a robot and any other agent located within an (approximately) circular range, as shown in Figure 7.2. On the contrary, in *case B*, the stereo vision system is employed to recognize and to measure the relative position between the robot's camera and any other agents located within its detection range as shown in 7.3.

5. All agents are equipped with a radio transceiver ensuring complete connectivity between pairs of robots as well as high-rate and low-latency communication. Long

Term Evolution (LTE) wireless modules can indeed meet such requirements [30].

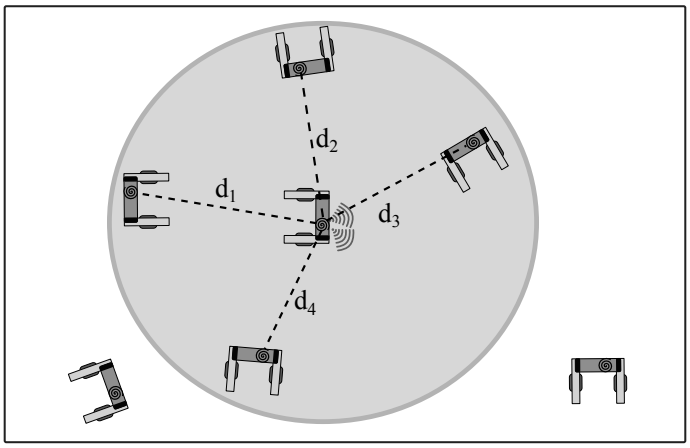


Figure 7.2: Qualitative overview of an omni-directional wireless ranging system mounted on each agent for distance measurement only

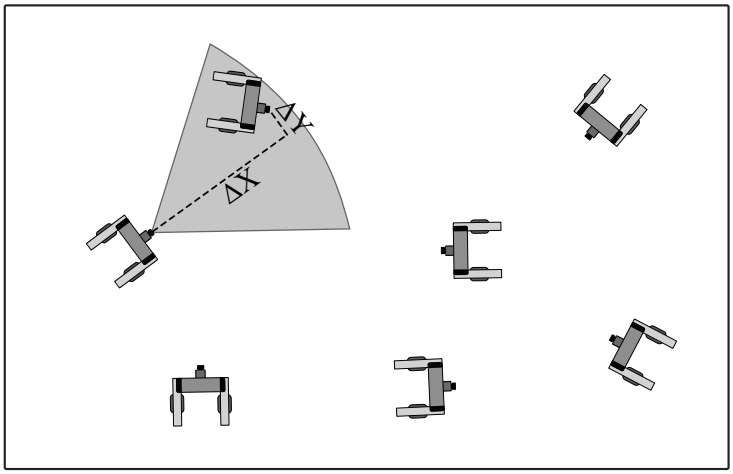


Figure 7.3: Qualitative overview of a front RGB-D camera measuring the relative position of two agents

Since the focus of this chapter is on Layer-2 localization, the local (layer-1) localization system has been simplified as follows:

- The effect of velocity drifts, i.e. μ and δ in (4.6), has been discarded. Thus, the system state has just three variables x,y,θ.
- The gyroscope has been removed from the sensing technologies, i.e. the layer-1 localization is performed by the fusion of odometry and QR code-based vision system.

Hence, the EKF for heading measurement is removed and the event-based local measurement model (4.17) becomes

$$\hat{\mathbf{o}}_k=\mathbf{h}(\mathbf{s}_k)+\boldsymbol{\zeta}_k=\begin{cases}\begin{bmatrix}\widehat{\Delta x}^c_k\\ \widehat{\Delta y}^c_k\\ \widehat{\Delta\theta}^c_k\end{bmatrix}=\begin{bmatrix}\mathbf{h}^c(\mathbf{s}_k)\\ \theta_q-\theta_k\end{bmatrix}+\begin{bmatrix}\boldsymbol{\zeta}^c_k\\ \zeta^c_{\theta,k}\end{bmatrix} & \text{QR detected}\end{cases} \tag{7.1}$$

Therefore, if each robot is modeled as a unicycle-like vehicle, the overall state transition of all agents in the navigation reference frame is given by the following non-linear discrete-time system [72], i.e.

$$\mathbf{p}_{k+1}=\begin{bmatrix}\mathbf{p}^{(1)}_{k+1}\\ \vdots\\ \mathbf{p}^{(N)}_{k+1}\end{bmatrix}=\begin{bmatrix}\mathbf{f}(\mathbf{p}^{(1)}_k,\mathbf{u}^{(1)}_k,\boldsymbol{\varepsilon}^{(1)}_k)\\ \vdots\\ \mathbf{f}(\mathbf{p}^{(N)}_k,\mathbf{u}^{(N)}_k,\boldsymbol{\varepsilon}^{(N)}_k)\end{bmatrix} \tag{7.2}$$

where

$$\mathbf{p}^{(i)}_{k+1}=\mathbf{f}(\mathbf{p}^{(i)}_k,\mathbf{u}^{(i)}_k,\boldsymbol{\varepsilon}^{(i)}_k)=\mathbf{f}(\mathbf{p}^{(i)}_k,\boldsymbol{\Delta\Phi}^{(i)}_k,\boldsymbol{\varepsilon}^{(i)}_k)\quad i=1,\ldots,N, \tag{7.3}$$

describes the process model of the $i-$th robot at time kT_s explained in (4.9) supposing that μ and δ in (4.9) are zero.

The observation model associated with system (7.2) includes two types of output functions, i.e.

- the geometrical relationship between the position and heading of each agent and those of any detected visual landmark in the same reference frame;

- the geometrical relationship between the pose of each robot and those of the other $N-1$ agents in the room.

As a result, the overall observation equation at time kT_s becomes

$$\mathbf{z}_k=\begin{bmatrix}\mathbf{z}^{(1)}_k\\ \vdots\\ \mathbf{z}^{(N)}_k\end{bmatrix}=\begin{bmatrix}\tilde{\mathbf{h}}^{(1)}(\mathbf{p}_k)\\ \vdots\\ \tilde{\mathbf{h}}^{(N)}(\mathbf{p}_k)\end{bmatrix}+\begin{bmatrix}\boldsymbol{\eta}^{(1)}_k\\ \vdots\\ \boldsymbol{\eta}^{(N)}_k\end{bmatrix}, \tag{7.4}$$

where vector $\mathbf{z}^{(i)}_k=\left[\mathbf{z}^{(i,1)}_k,\ldots,\mathbf{z}^{(i,N)}_k\right]^T$ includes all possible observations from agent i, $\boldsymbol{\eta}^{(i)}_k=\left[\boldsymbol{\eta}^{(i,1)}_k,\ldots,\boldsymbol{\eta}^{(i,N)}_k\right]^T$ is the vector comprising the respective measurement uncertainty

contributions, and, finally,

$$\tilde{\mathbf{h}}^{(i)}(\mathbf{p}_k)=\begin{bmatrix} \mathbf{h}(\mathbf{p}_k^{(i)},\mathbf{p}_k^{(1)}) \\ \vdots \\ \mathbf{h}(\mathbf{p}_k^{(i)},\mathbf{p}_k^{(i-1)}) \\ \boldsymbol{o}(\mathbf{p}_k^{(i)}) \\ \mathbf{h}(\mathbf{p}_k^{(i)},\mathbf{p}_k^{(i+1)}) \\ \vdots \\ \mathbf{h}(\mathbf{p}_k^{(i)},\mathbf{p}_k^{(N)}) \end{bmatrix}, \quad i=1,\ldots,N, \tag{7.5}$$

is the vector including all observations from agent i. Note that the $i-$th function of the vector, referred to as $\boldsymbol{o}(\cdot)$, is the ith agent local (layer 1) observation model as (7.1). Each function $\mathbf{h}(\cdot,\cdot)$ consists of M equations and depends on how the state variables of agents $j=1,\ldots,N$ for $j\neq i$ are actually observed by the $i-$th robot. Hence, the equations of $\mathbf{h}(\cdot,\cdot)$ differ in *case A* and *case B*, respectively, in accordance with assumption 4.

7.2 Position estimation algorithm

The proposed estimation technique is an interlaced extended kalman filter (IEKF) based on the process and measurement models described above. The state evolution in the process model of each agent does not depend on the state of the other robots. Therefore, the Kalman filtering *prediction step* is similar to the individual agent's prediction precess explained in previous chapters. The only difference in the prediction is that we now have N independent standard EKFs.

$$\begin{aligned} \hat{\mathbf{p}}_{k+1|k}^{(i)} &= \mathbf{f}(\hat{\mathbf{p}}_{k|k}^{(i)}, \widehat{\boldsymbol{\Delta\Phi}}_k^{(i)}) \\ P_{k+1|k}^{(i)} &= F_k^{(i)} P_{k|k}^{(i)} F_k^{(i)^T} + G_k^{(i)} Q_k^{(i)} G_k^{(i)^T} \end{aligned} \quad i=1,\ldots,N \tag{7.6}$$

Observe that (7.6) depends just on local quantities. Therefore, the prediction step equations can be computed locally, i.e. on board of each robot, thus ensuring a fully distributed implementation.

The update step takes in to account the new event-based measurements depending on which agents can connect and detect each other. Due to the definition of the measurement model (7.4), the updated state estimate of the $i-$th agent and its covariance matrix depend not only on the respective predicted values and on the measurement data, but also on the predicted state and on the predicted covariance of the other agents. Nevertheless, generally an agent is not able to observe all the other agents simultaneously. This means that all observations are inherently intermittent, as they depend on the reading range of

the chosen measurement systems and on the distance between each robot and both the other agents and one of the landmarks. As a result of all the issues above, the *update step* of the proposed IEKF is inherently stochastic and event-based in the case considered.

In the IEKF described in [72], the computation of the innovation term associated with a generic pair of agents (i, j) relies on the predicted state of j regarded as an additional measure. Therefore, both $\hat{\mathbf{p}}^{(j)}_{k+1|k}$ and its covariance matrix $P^{(j)}_{k+1|k}$ have to be transmitted to agent i, thus "interlacing" the two subsystems. In particular, $P^{(j)}_{k+1|k}$ has to be included in the Kalman gain, as it will be shown in the following, to keep into account the fact that $\hat{\mathbf{p}}^{(j)}_{k+1|k}$ is affected by some uncertainty.

By extending a similar approach of EKF with event-based measurements to the system at hand for all pairs of agents i and j, we can define a binary random variable $\tau^{(i,j)}_k$, to be set to 1 if agent i is able to observe agent j at time kT_s, or 0 otherwise. Similarly, $\tau^{(i,i)}_k$ is set to 1 if robot i is able to detect a landmark at time kT_s, or 0 otherwise. Starting from the basic *update step* equations of an EKF and assuming to replace the variance of real measurements with a large dummy value anytime $\tau^{(i,j)}_k = 0$, after some algebraic steps it can be shown that, if the dummy variance tends to infinity [96], then the update equations of the IEKF running on agent i become

$$\begin{aligned} \hat{\mathbf{p}}^{(i)}_{k+1|k+1} &= \hat{\mathbf{p}}^{(i)}_{k+1|k} + K^{(i)}_{k+1} T^{(i)}_{k+1} [\mathbf{z}^{(i)}_{k+1} - \tilde{\mathbf{h}}^{(i)}(\hat{\mathbf{p}}_{k+1|k})] \\ P^{(i)}_{k+1|k+1} &= P^{(i)}_{k+1|k} - K^{(i)}_{k+1} T^{(i)}_{k+1} \tilde{H}^{(i,i)}_{k+1} P^{(i)}_{k+1|k} \end{aligned} \tag{7.7}$$

where

$$T^{(i)}_{k+1} = \begin{bmatrix} \tau^{(i,1)}_{k+1} I_M & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \tau^{(i,i)}_{k+1} I_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & \tau^{(i,N)}_{k+1} I_M \end{bmatrix} \tag{7.8}$$

is an $M \cdot (N-1)+3 \times M \cdot (N-1)+3$ diagonal matrix made of binary random variables (as all observations can be reasonably assumed to be independent) and I_M is the Identity matrix of size M. Besides, $\tilde{H}^{(i,i)}_{k+1}$ is the Jacobian of $\tilde{\mathbf{h}}^{(i)}(\cdot)$ with respect to $\mathbf{p}^{(i)}$ computed at $\hat{\mathbf{p}}_{k+1|k}$ and

$$K^{(i)}_{k+1} = P^{(i)}_{k+1|k} \tilde{H}^{(i,i)^T}_{k+1} \left[\tilde{H}^{(i,i)}_{k+1} P^{(i)}_{k+1|k} \tilde{H}^{(i,i)^T}_{k+1} + \tilde{S}^{(i)}_{k+1} + \tilde{R}^{(i)}_{k+1} \right]^{-1} \tag{7.9}$$

is the Kalman gain of the IEKF running on the $i-$th agent. Observe that (7.9) comprises two measurement covariance matrices instead of just one, i.e. the block diagonal

covariance matrix

$$\tilde{R}_{k+1}^{(i)} = \begin{bmatrix} R_{k+1}^{(i,1)} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ \cdots & \cdots & R_{k+1}^{(i,i)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & R_{k+1}^{(i,N)} \end{bmatrix} \tag{7.10}$$

and

$$\tilde{S}_{k+1}^{(i)} = \sum_{j=1 \wedge j \neq i}^{N} \tilde{H}_{k+1}^{(i,j)} P_{k+1|k}^{(j)} \tilde{H}_{k+1}^{(i,j)^T}. \tag{7.11}$$

Matrix (7.10) includes both the covariance matrix $R_{k+1}^{(i,j)}$ associated with the relative pose measurements between each pair of agents (i, j) and the covariance matrix $R_{k+1}^{(i,i)}$ due to the absolute position and heading measures injected into the IEKF anytime a local localization measurement is available, i.e. a QR code is detected by $i-th$ agent. Matrix (7.11) instead takes into account the covariances $P_{k+1|k}^{(j)}$ of the states predicted by the agents different from i, with $\tilde{H}_{k+1}^{(i,j)}$ being the Jacobian of (7.5) with respect to $\mathbf{p}^{(j)}$ for $i \neq j$ and computed at $\hat{\mathbf{p}}_{k+1|k}$.

It is worth emphasizing that expressions (7.6) and (7.7) are absolutely general; however, the actual implementation of (7.7) depends on the observation model used which differs according to the sensing technology used to measure the relative location of an agent with respect to the others. As introduced in Section 7.1, we use two different measurement sources and compare them in this chapter.

***Case A*: Omni-directional wireless ranging system**

As shown in Figure 7.2, a wireless ranging system can cover a large area. Such system can measure the distance from its corresponding agent to any other agent which is located inside its detection area. In this case

$$\tilde{\mathbf{h}}^{(i)}(\mathbf{p}_k) = \tilde{\mathbf{h}}_A^{(i)}(\mathbf{p}_k) = [\mathbf{h}_A(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(1)}), \ldots, \boldsymbol{o}(\mathbf{p}_k^{(i)}), \ldots, \mathbf{h}_A(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(N)})]^T \tag{7.12}$$

where

$$\mathbf{h}_A(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(j)}) = \sqrt{(x_k^{(j)} - x_k^{(i)})^2 + (y_k^{(j)} - y_k^{(i)})^2} \tag{7.13}$$

and $o(\mathbf{p}_k^{(i)}) = o_k^{(i)}$ is as (7.1).

***Case B*: RGB-D front camera system**

The RGB-D front cameras, like Kinect, can capture the area in front of them, detect another agent and measure the planar 2-dimensional distance between two agents in the

camera frame. The process is very similar to how the camera can measure the Δx^c and Δy^c distance to the QR code as explained in Chapter 4. Supposing that $\Delta x_k^{i,j}$ and $\Delta y_k^{i,j}$ are the measured distance from the $i-th$ to $j-th$ agent in the $x-axis$ and $y-axis$ direction of the camera frame respectively, we have

$$\tilde{\mathbf{h}}^{(i)}(\mathbf{p}_k) = \tilde{\mathbf{h}}_B^{(i)}(\mathbf{p}_k) = [\mathbf{h}_B(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(1)}), \ldots, \boldsymbol{o}(\mathbf{p}_k^{(i)}), \ldots, \mathbf{h}_B(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(N)})]^T \tag{7.14}$$

with

$$\mathbf{h}_B(\mathbf{p}_k^{(i)}, \mathbf{p}_k^{(j)}) = \begin{bmatrix} (x_k^{(j)} - x_k^{(i)}) \cos\theta_k^{(i)} + (y_k^{(j)} - y_k^{(i)}) \sin\theta_k^{(i)} \\ -(x_k^{(j)} - x_k^{(i)}) \sin\theta_k^{(i)} + (y_k^{(j)} - y_k^{(i)}) \cos\theta_k^{(i)} \end{bmatrix} \tag{7.15}$$

Of course, a similar notation can be extended to all variables used in expressions (7.6)-(7.11). For instance, $R_{A,k+1}^{(i,j)}$ and $R_{B,k+1}^{(i,j)}$ denote the covariance matrices associated with the relative measurements between agents i and j in *case A* and *case B*, respectively, at time $(k+1)T_s$. Since such matrices can be assumed to be stationary, the time index can be omitted in the following. Notice that both (7.13) and (7.15) are nonlinear, but the number of observations M when another agent is detected is different in either case. In *case A*, $M = 1$ since low-cost wireless ranging systems can just measure the distance between transmitter and receiver, whereas in *case B*, $M = 2$ because the relative offsets between agents i and j along axes x and y can be easily extracted from the image collected by the RGB-D camera. On the contrary, measuring the relative orientation between i and j requires more sophisticated image processing algorithms and it is a measurement problem on its own. This is why this kind of measurements has not been included in the present analysis.

A final remark is on communication latency, which may influence measurement uncertainty significantly, due to the difference between the time when the predicted state of agent j is sent to agent i and the moment when the relative pose of j is actually measured by i. If assumption 5 defined in Section 7.1 holds, then the impact of communication latencies is negligible, provided that robots move so slowly that their linear and angular displacements during the time interval spent for communication is reasonably small. On the contrary, if the communication latency becomes significant, then the uncertainty contributions affecting all measurements of position, distance and heading should be properly estimated and used to boost the elements of $R_A^{(i,j)}$ and $R_B^{(i,j)}$.

7.3 Simulation results

In order to evaluate the positioning accuracy with and without collaborative localization in *case A* and *B*, respectively, the results of some Monte Carlo simulations in different conditions are reported in the following. The main simulation parameters are listed below:

- Duration of each simulated path: about 120 s;
- Number N of agents in the room: between 2 and 10;
- Room size: 100 m^2;
- Number of random paths of each robot: 24;
- Robots linear velocity range: $[0, 2]$ m/s (at such speeds assumption 5 holds, so the effect of communication latencies can be assumed to be negligible);
- Robots angular velocity range: $[-\frac{\pi}{2}, \frac{\pi}{2}]$ rad/s;
- Odometers sampling period: $T_s = 4$ ms;
- The uncertainty associated with the encoders and the camera for QR code reading are the same as determined and used in the previous chapters.
- Distance between QR codes: 2 m;
- Wireless system detection range: about 10 m;
- Rate of wireless range measurements: about 25 Hz;
- Variance of indoor wireless distance measurement data anytime agent j is detected: $R_A^{(i,j)} \approx 0.45$ m^2, in accordance with the experimental results published in the scientific literature [80, 59];
- RGB-D camera reading range (according to Kinect-like specifications): from 0.8 m to 3.5 m with a horizontal aperture angle of about 57°;
- Covariance matrix of stereo camera measurements anytime agent j is detected (based on Kinect's v.2 average accuracy reported in [24]): $R_B^{(i,j)} = \mathrm{diag}(6.1{\cdot}10^{-7} d^{(i,j)^2}\mathrm{m}^2,\ 6.25{\cdot}10^{-6}\mathrm{m}^2)$, with $d^{(i,j)}$ being the distance between agents i and j along the focal axis of the camera.
- Camera and Kinect image acquisition rate: about 10 Hz.

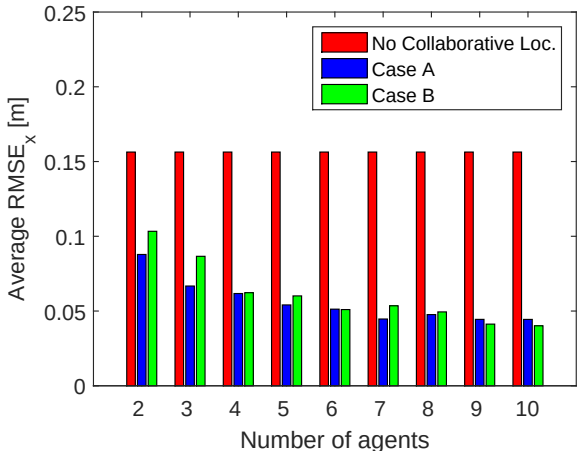
Tab. I shows the simulated probabilities (expressed in %) that agent i detects some other agent in *Case A* and *B*, respectively. The reported values keep into consideration the actual data rates of the sensors employed for collaborative localization. Given that the RGB-D image acquisition and processing rate is notoriously quite low and its reading range is also much more limited than the range of a wireless system, the probability of agent detection in *Case B* is about 10 times smaller than in *Case A*. Observe that while

in *Case B* the detection probability slightly grows with the number of agents, in *Case A* it saturates to about 10%. This is reasonable, because, even though the size of the room is smaller than the nominal wireless reading range, the rate of wireless distance measurements is 10 times smaller than the system sampling frequency. Thus, only once out of 10 times at least one of binary variables $\tau^{(i,j)}$ is equal to 1 for $i \neq j$.

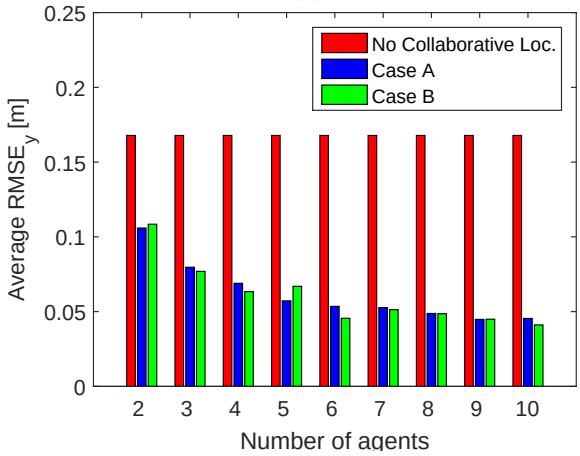
Table 7.1: Relative frequency of detection (expressed in %) of some other agents in *Case A* and in *Case B*, respectively.

No. Agents	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Case A	9.8	9.9	10	10	10	10	10	10	10
Case B	0.2	0.4	0.5	0.7	0.9	1.0	1.2	1.3	1.4

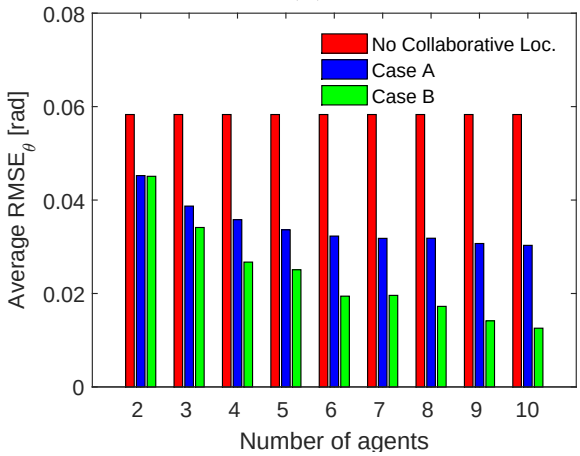
Figure 7.4 shows the average root mean square estimation errors (RMSE) of state variables x (a), y (b) and θ (c) of one of the agents as a function of the number of robots N moving freely in the room. First, the RMSE values for each path are computed. Then, they are averaged together. Different bars refer to *Case A*, *Case B* and without collaborative localization, respectively. Notice that the results without collaborative localization do not depend on N, as expected, because the same 24 paths are used in all tests. On the other hand, both collaborative localization strategies greatly enhance the accuracy in estimating the state variables. Of course, such an improvement is more evident when the number of agents grows, due to the availability of a larger amount of relative measurement data from nearby robots. Quite interestingly, in spite of some fluctuations due to the limited set of simulated paths, the estimation accuracy of x and y in *Cases A*, *B* is comparable for a given N. The greater amount of available information due to both larger wireless coverage and higher data rate indeed compensate the poorer accuracy of wireless ranging. As far as state variable θ is concerned, estimation accuracy is generally clearly better in *Case B*. Also, the accuracy gap compared to *Case A* tends to grow with N, because wireless ranging cannot measure the relative position of two agents in a 2D reference frame.



(a)



(b)



(c)

Figure 7.4: Average RMS estimation errors of state variables x (a), y (b) and θ (c) for one of the agents as a function of the number of robots in the room and in *case A*, *case B* and without collaborative localization, respectively.

Chapter 8

Conclusion

This dissertation proposes affordable localization techniques for wheeled robots moving in large indoor environments. As introduced in Chapter 1, the main contributions of this dissertation cover three complimentary subjects.

8.1 Self-localization

The approach described in this dissertation is based on multi-sensor data fusion techniques integrating absolute and relative position measurement technologies. For large environments, the systems detecting landmarks or reference points stuck on the floor are used for absolute position measurement. Hence, a low cost RGB-D camera is used to detect QR codes placed on the floor at known positions. The sporadic event-based camera measurements are fused with odometry and IMU data using various recursive estimators. A comparative performance analysis based on experiments conducted on the field show not only that the proposed idea provides an acceptable trade-off between accuracy, complexity, cost and robustness, but also that the EHF provides the most accurate estimation for the problem at hand.

A major issue of using event-based measurements is the switching measurement model which leads to temporary lack of observability. This brings about some correlations between the state variables which sometimes results in noticeable updates of the position variables (i.e. x and y) while they are not observable. The observability problem should be studied more in depth. Maybe some changes in the system model can partially overcome this problem. For instance, if the IMU outcome is regarded as an input rather than an output where the angular velocity is estimated as a Bayesian fusion of encoder and gyroscope measurements, the filters update step will be executed only when a QR code is detected. In this situation, both position and heading are observable.

Moreover, the proposed localization technique claims that it is appropriate for wheeled

robots regardless of the room size and crowd amount. The technique has been tested in environments of different size. In all cases, the environment size did not significantly influence the results. However, due to lack of resources, no test was conducted in crowded environments.

The vision system also needs some further studies, e.g. to include measurement and computational delays as the new state variables in the model and adding some techniques to observe them.

The results of the proposed estimators are shown in Chapter 5. A summary and possible future directions for the proposed estimators are described in the following.

EKF

The EKF has been designed using the real sensors characteristics. In the future, one may like to test the second-order EKF or some kinds of adaptive Kalman filters. However, the adaptive filters should be designed with significant care since improper parameter adaptation may result in filter divergence.

UKF

In this dissertation, UKF has proved to converge to the steady state much faster than the other filters. However, the computational cost is a problem. Besides, it does not offer significant benefits in terms of accuracy compared with the EKF. Nonetheless, in the experiments, the UKF scaling parameters, i.e. α, β and κ were fixed and derived from the literature. For the problem at hand, the values of such parameters could be further optimized.

PF

Particle filters include a large family of estimators and selecting the best one for the current problem can be a research project itself. The major issue in Particle filters is the high computational cost which is mostly due to the large number of particles and the resampling procedure. Other kinds of particle filtering may be able to reduce the required number of particles and hence, decrease the computational time.

EHF

It has been shown that for the system at hand, EHF is noticeably the most accurate estimator. However, the parameters chosen for the EHF may affect filter robustness. This problem should be better analyzed in the future in order to achieve a better trade-off between accuracy and robustness.

8.2 Optimal Placement

In Chapter 6, an optimal placement strategy of landmarks for indoor localization is presented. In particular, assuming to have i) a regular triangular lattice of landmarks in a wide-open room and ii) a sensor with a limited detection area (approximated as a triangle, as well), the optimal distance between pairs of adjacent landmarks was derived both numerically and analytically. Both solutions converge to the same results. The analytical solution is particularly valuable because it provides a very simple, closed-form expression, which depends just on the radial and angular detection ranges of the sensor adopted, regardless of the specific sensing technology. Therefore, it can be applied in a multitude of contexts.

The results of several Monte Carlo simulations confirm that, if an optimal layout for a given type of sensors is deployed in a wide-open room without obstacles, localization accuracy is limited mainly by the uncertainty associated with the measurement of position and orientation between sensor and landmarks. Of course, in real indoor environments, where also walls and obstacles are present, a perfect optimal layout can be hardly deployed. As a consequence, just suboptimal results can be achieved, either because some landmarks could be difficult to detect or because they could not be placed at all. In such conditions, the longer the sensor detection range is, the higher the probability of missing some landmark becomes, thus potentially degrading localization accuracy.

The results of this part of research can pave the way to various future activities. First of all, other kinds of lattices, e.g. squares and hexagons should be studied. Then, more realistic sensor detection areas should be taken into account. In fact, sensors may have other detection area shapes, e.g. trapezoidal or cone shape. Another issue to be addressed is that some sensors may need to detect more than one landmark at the same time. Thus, for such sensors, the solution should be extended in order to ensure detection of at least n landmarks instead of one.

The problem of landmarks placement in real environments paves the way to further research on placement optimization, which should take into account not only the geometry of the environment, but also the probabilities of transition between pairs of adjacent landmarks. In order to address this problem, a lattice of landmarks could be turned into a Markov chain model, in which nodes and edges of the graph represent, respectively, the available landmarks and the transitions between pairs of them (with a given probability) because of the motion of the agent to be localized. The main challenge of using this model is that the transition probabilities depend on both the real users paths and the constraints imposed by the environment. On the other hand, a considerable advantage of using this approach is that those landmarks that are unlikely to be detected or encountered (thus

marginally affecting localization accuracy) could be easily identified. As a consequence, it would be possible to find the minimum number of landmarks that ensure a localization uncertainty below a certain threshold with a given level of confidence.

8.3 Collaborative Localization

Collaborative or synergic localization refers to the ability of a group of robots to refine their own estimated positions by using both neighbors' states and relative measurements of distance and/or position. In this dissertation, the performances of two alternative collaborative localization strategies, both based on a common underlying Interlaced Extended Kalman Filter (IEKF), were compared through simulations. The reported results confirm that the effectiveness of collaborative localization becomes more evident when the probability of detecting other agents in the environment grows. In the two cases considered, the use of an RGB-D camera seems to be globally preferable to wireless ranging, although the difference, in terms of accuracy, was not so impressive as it was expected.

Future work could be focused on a more in-depth analysis of the trade-off between these two scenarios, possibly finding an analytical relationship between detection probability, measurement accuracy and target performance. Another interesting research work which can be taken in to account is to define and investigate the possibility of using an Interlaced H_∞ extended Kalman filter.

Bibliography

[1] Leopoldo Angrisani, Aldo Baccigalupi, and Rosario Schiano Lo Moriello. A measurement method based on kalman filtering for ultrasonic time-of-flight estimation. *IEEE Transactions on Instrumentation and Measurement*, 55(2):442–448, 2006.

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[3] A. Bahr, M.R. Walter, and J.J. Leonard. Consistent cooperative localization. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 3415–3422, Kobe, Japan, May 2009.

[4] Tim Bailey, M. Bryson, Hua Mu, J. Vial, L. McCalman, and H. Durrant-Whyte. Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 2859–2865, Shanghai, China, May 2011.

[5] Walter William Rouse Ball. *Mathematical recreations and essays*. MacMillan, 1914.

[6] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[7] Anahid Basiri, Pouria Amirian, and Adam Winstanley. The use of quick response (qr) codes in landmark-based pedestrian navigation. *International Journal of Navigation and Observation*, 2014, 2014.

[8] Anahid Basiri, Pouria Amirian, and Adam Winstanley. The use of quick response (QR) codes in landmark-based pedestrian navigation. *International Journal of Navigation and Observation*, 2014:1–7, Apr. 2014.

[9] Maximilian Beinhofer, Jörg Müller, and Wolfram Burgard. Effective landmark placement for accurate and reliable mobile robot navigation. *ROBOT. AUTON. SYST.*, 61(10):1060–1069, Oct. 2013.

[10] Maximilian Beinhofer, Jörg Müller, Andreas Krause, and Wolfram Burgard. Robust landmark selection for mobile robot navigation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3637–2643. IEEE, 2013.

[11] Luiz Belussi and Nina Hirata. Fast qr code detection in arbitrarily acquired images. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 281–288. IEEE, 2011.

[12] Michael Bloesch, Sammy Omari, Péter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A Hoepflinger, Marco Hutter, and Roland Siegwart. Fusion of optical flow and inertial measurements for robust egomotion estimation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3102–3107. IEEE, 2014.

[13] Andrea Bonarini, Matteo Matteucci, and Marcello Restelli. Automatic error detection and reduction for an odometric sensor based on two optical mice. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 1675–1680. IEEE, 2005.

[14] Johann Borenstein, Liqiang Feng, and HR Everett. *Navigating mobile robots: Systems and techniques*. AK Peters, Ltd., 1996.

[15] R.M. Buehrer, Tao Jia, and B. Thompson. Cooperative indoor position location using the parallel projection method. In *Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, Zurich, Switzerland, Sep. 2010.

[16] Byoung-Suk Choi, Joon-Woo Lee, Ju-Jang Lee, and Kyoung-Taik Park. A hierarchical algorithm for indoor mobile robot localization using RFID sensor fusion. *IEEE Trans. on Industrial Electronics*, 58(6):2226–2235, Jun. 2011.

[17] Byoung-Suk Choi and Ju-Jang Lee. Sensor network based localization algorithm using fusion sensor-agent for indoor service robot. *IEEE Trans. on Consumer Electronics*, 56(3):1457–1465, Aug. 2010.

[18] Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects. *Information fusion*, 7(2):221–230, 2006.

[19] Yingying Chen, John-Austen Francisco, Wade Trappe, and Richard P Martin. A practical approach to landmark deployment for indoor localization. In *IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Reston, VA, USA, Sep. 2006.

[20] Yang Cheng, Mark W Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers-a tool to ensure accurate driving and science imaging. *IEEE Robotics & Automation Magazine*, 13(2):54–62, 2006.

[21] Hakyoung Chung, L. Ojeda, and J. Borenstein. Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. *IEEE Transactions on Robotics and Automation*, 17(1):80–84, Feb. 2001.

[22] A Colombo, D Fontanelli, D Macii, and L Palopoli. A wearable embedded inertial platform with wireless connectivity for indoor position tracking. In *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*, pages 1–6. IEEE, 2011.

[23] Alessio Colombo, Daniele Fontanelli, David Macii, and Luigi Palopoli. Flexible indoor localization and tracking based on a wearable platform and sensor data fusion. *IEEE Transactions on Instrumentation and Measurement*, 63(4):864–876, 2014.

[24] Andrea Corti, Silvio Giancola, Giacomo Mainetti, and Remo Sala. A metrological characterization of the kinect V2 time-of-flight camera. *Robotics and Autonomous Systems*, 75, Part B(1):584 – 594, Jan. 2016.

[25] Mariolino De Cecco. Sensor fusion of inertial-odometric navigation as a function of the actual manoeuvres of autonomous guided vehicles. *Measurement Science and Technology*, 14(5):643, 2003.

[26] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.

[27] Emidio DiGiampaolo and Francesco Martinelli. Mobile robot localization using the phase of passive uhf rfid signals. *IEEE Transactions on Industrial Electronics*, 61(1):365–376, 2014.

[28] John C Doyle, Keith Glover, Pramod P Khargonekar, and Bruce A Francis. State-space solutions to standard h 2 and h control problems. *IEEE Transactions on Automatic control*, 34(8):831–847, 1989.

[29] Borislav Dzodzo, Long Han, Xu Chen, Huihuan Qian, and Yangsheng Xu. Realtime 2d code based localization for indoor robot navigation. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 486–492. IEEE, 2013.

[30] A. Elnashar and M.A. El-Saidny. Looking at LTE in practice: A performance analysis of the LTE system based on field test results. *IEEE Vehicular Technology Magazine*, 8(3):81–92, Sep. 2013.

[31] Lawrence H Erickson and Steven M LaValle. *An art gallery approach to ensuring that landmarks are distinguishable*, pages 81–88. MIT Press, 2012.

[32] Jan Figat and Włodzimierz Kasprzak. Nao-mark vs qr-code recognition by nao robot vision. In *Progress in Automation, Robotics and Measuring Techniques*, pages 55–64. Springer, 2015.

[33] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.

[34] S. Frattasi, M. Monti, and R. Prasad. The synergic localization system (SLS). In *Proc. Vehicular Technology Conference (VTC)*, volume 4, pages 2753–2755, Dallas, TX, USA, Sep. 2005.

[35] Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual slam. *IEEE Transactions on Robotics*, 24(5):1054–1065, 2008.

[36] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[37] Larry W Fullerton, James L Richards, and Ivan A Cowie. System and method for position determination by impulse radio using round trip time-of-flight, August 26 2003. US Patent 6,611,234.

[38] Cimini Gionata, Ferracuti Francesco, Freddi Alessandro, Iarlori Sabrina, and Monteri Andrea. An inertial and QR code landmarks-based navigation system for impaired wheelchair users. In Sauro Longhi, Pietro Siciliano, Michele Germani, and Andrea Monteri, editors, *Ambient Assisted Living*, pages 205–214. Springer International Publishing, 2014.

[39] P Goel, SI Roumeliotis, and GS Sukhatme. Robot localization using relative and absolute position estimates in proc ieee int. In *Conference on Robots and Systems, October, Kyongju, Korea*, 1999.

[40] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, volume 140, pages 107–113. IET, 1993.

[41] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.

[42] Ismail Guvenc and Chia-Chin Chong. A survey on toa based wireless localization and nlos mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3):107–124, 2009.

[43] Babak Hassibi, Thomas Kailath, and Ali H Sayed. Array algorithms for h estimation. *IEEE Transactions on Automatic Control*, 45(4):702–706, 2000.

[44] Dirk Helbing and Peter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E Stat. Nonlin. Soft.*, 51:4282–4286, May 1995.

[45] Hwan Hur and Hyo-Sung Ahn. Unknown input h_∞ observer-based localization of a mobile robot with sensor failure. *IEEE/ASME Transactions on Mechatronics*, 19(6):1830–1838, 2014.

[46] F. Ijaz, Hee Kwon Yang, A.W. Ahmad, and Chankil Lee. Indoor positioning: A review of indoor ultrasonic positioning systems. In *15th International Conference on Advanced Communication Technology (ICACT)*, pages 1146–1150, Jan. 2013.

[47] Christopher Jekeli. *Inertial navigation systems with geodetic applications*. Walter de Gruyter, 2001.

[48] Mustafa Ozgur Kanli et al. Limitations of pseudolite systems using off-the-shelf gps receivers. *Positioning*, 1(08), 2004.

[49] A. Kealy, G. Retscher, N. Alam, A. Hasnur-Rabiain, C. Toth, D.A. Grejner-Brzezinska, T. Moore, C. Hill, V. Gikas, C. Hide, C. Danezis, L. Bonenberg, and G.W. Roberts. Collaborative navigation with ground vehicles and personal navigators. In *Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sydney, Australia, Nov. 2012.

[50] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[51] A. A. Khaliq, F. Pecora, and A. Saffiotti. Inexpensive, reliable and localization-free navigation using an RFID floor. In *European Conf. on Mobile Robots (ECMR)*, pages 1–7, Lincoln, United Kingdom, Sep. 2015.

[52] J. Knuth and P. Barooah. Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 1534–1539, Karlsruhe, Germany, May 2013.

[53] Hakan Koyuncu and Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.

[54] Paweł Kułakowski, Javier Vales-Alonso, Esteban Egea-López, Wiesław Ludwin, and Joan García-Haro. Angle-of-arrival localization based on antenna arrays for wireless sensor networks. *Computers & Electrical Engineering*, 36(6):1181–1186, 2010.

[55] Vladimir Kulyukin, Aliasgar Kutiyanawala, Edmund LoPresti, Judith Matthews, and Richard Simpson. iwalker: Toward a rollator-mounted wayfinding system for the elderly. In *2008 IEEE International Conference on RFID*, pages 303–311. IEEE, 2008.

[56] Tiancheng Li, Miodrag Bolic, and Petar M Djuric. Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal processing magazine*, 32(3):70–86, 2015.

[57] Wenling Li and Yingmin Jia. H-infinity filtering for a class of nonlinear discrete-time systems based on unscented transform. *Signal Processing*, 90(12):3301–3307, 2010.

[58] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[59] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli. A data fusion technique for wireless ranging performance improvement. *IEEE Transactions on Instrumentation and Measurement*, 62(1):27–37, Jan. 2013.

[60] David Macii, Andrea Boni, Mariolino De Cecco, and Dario Petri. Tutorial 14: Multisensor data fusion. *IEEE Instrumentation & Measurement Magazine*, 11(3):24–33, 2008.

[61] Krishan M Nagpal and Pramod P Khargonekar. Filtering and smoothing in an h setting. *IEEE Transactions on Automatic Control*, 36(2):152–166, 1991.

[62] United Nations. World population ageing: 1950-2050. *Department of Economic and Social*, 2002.

[63] Payam Nazemzadeh, Daniele Fontanelli, and David Macii. An indoor position tracking technique based on data fusion for ambient assisted living. In *2013 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 7–12. IEEE, 2013.

[64] Payam Nazemzadeh, Daniele Fontanelli, David Macii, Tizar Rizano, and Luigi Palopoli. Design and performance analysis of an indoor position tracking technique for smart rollators. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[65] Payam Nazemzadeh, Federico Moro, Daniele Fontanelli, David Macii, and Luigi Palopoli. Indoor positioning of a robotic walking assistant for large public environments. *TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 2015.

[66] Igor Neri, Riccardo Centonze, Mario Luca Fravolini, and Antonio Moschitta. A simple ranging technique based on received signal strength measurements in a narrowband 2.4 ghz channel: A space diversity approach. In *Measurements and Networking Proceedings (M&N), 2013 IEEE International Workshop on*, pages 189–194. IEEE, 2013.

[67] John-Olof Nilsson. Infrastructure-free pedestrian localization. 2013.

[68] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE, 2004.

[69] I Ohya, Akio Kosaka, and Avi Kak. Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*, 14(6):969–978, 1998.

[70] Luigi Palopoli, Antonis Argyros, Josef Birchbauer, Alessio Colombo, Daniele Fontanelli, Axel Legay, Andrea Garulli, Antonello Giannitrapani, David Macii, Federico Moro, et al. Navigation assistance and guidance of older adults across complex public spaces: the dali approach. *Intelligent Service Robotics*, 8(2):77–92, 2015.

[71] Ghazaleh Panahandeh and Magnus Jansson. Vision-aided inertial navigation based on ground plane feature detection. *IEEE/ASME transactions on mechatronics*, 19(4):1206–1215, 2014.

[72] Stefano Panzieri, F. Pascucci, and Roberto Setola. Multirobot localisation using interlaced extended Kalman filter. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2816–2821, Beijing, China, Oct. 2006.

[73] Stefano Panzieri, Federica Pascucci, Lorenzo Sciavicco, and Roberto Setola. Distributed cooperative localization. *Journal of Information Technology Research*, 6(3):49–67, Jul. 2013.

[74] Sunhong Park and Shuji Hashimoto. Autonomous mobile robot navigation using passive rfid in indoor environment. *IEEE Transactions on Industrial Electronics*, 56(7):2366–2373, 2009.

[75] Ryan Parker and Shahrokh Valaee. Vehicular node localization using received-signal-strength indicator. *IEEE Transactions on Vehicular Technology*, 56(6):3371–3380, 2007.

[76] Valter Pasku, Alessio De Angelis, Guido De Angelis, Antonio Moschitta, and Paolo Carbone. Magnetic field analysis for distance measurement in 3d positioning applications. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6. IEEE, 2016.

[77] Valter Pasku, Alessio De Angelis, Marco Dionigi, Guido De Angelis, Antonio Moschitta, and Paolo Carbone. A positioning system based on low frequency magnetic fields. *IEEE Transactions on Industrial Electronics*, 2015.

[78] Valter Pasku, Alessio De Angelis, Marco Dionigi, Antonio Moschitta, Guido De Angelis, and Paolo Carbone. Analysis of nonideal effects and performance in magnetic positioning systems. *IEEE Transactions on Instrumentation and Measurement*, PP(99):1–12, 2016.

[79] Valter Pasku, Alessio De Angelis, Antonio Moschitta, Paolo Carbone, John-Olof Nilsson, Satyam Dwivedi, and Peter Händel. A magnetic ranging aided dead-reckoning indoor positioning system for pedestrian applications. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6. IEEE, 2016.

[80] P. Pivato, S. Dalpez, and D. Macii. Performance evaluation of chirp spread spectrum ranging for indoor embedded navigation systems. In *Proc. of IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 307–310, Karlsruhe, Germany, June 2012.

[81] Paolo Pivato, Luigi Palopoli, and Dario Petri. Accuracy of rss-based centroid localization algorithms in an indoor environment. *IEEE Transactions on Instrumentation and Measurement*, 60(10):3451–3460, 2011.

[82] A. Prorok and A. Martinoli. A reciprocal sampling algorithm for lightweight distributed multi-robot localization. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3241–3247, San Francisco, CA, USA, Sep. 2011.

[83] Ioannis M Rekleitis. A particle filter tutorial for mobile robot localization. *Centre for Intelligent Machines, McGill University, Tech. Rep. TR-CIM-04-02*, 2004.

[84] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. Beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 19(7):37–38, 2004.

[85] S.I. Roumeliotis and George A. Bekey. Synergetic localization for groups of mobile robots. In *Proc. of IEEE Conference on Decision and Control*, volume 4, pages 3477–3482, Sydney, Australia, Dec. 2000.

[86] Angga Rusdinar, Jungmin Kim, Junha Lee, and Sungshin Kim. Implementation of real-time positioning system using extended kalman filter and artificial landmark on ceiling. *Journal of Mechanical Science and Technology*, 26(3):949–958, 2012.

[87] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson. Landmark selection for vision-based navigation. *IEEE Trans. Robot.*, 22(2):334–349, Apr. 2006.

[88] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

[89] S Schon and Olaf Bielenberg. On the capability of high sensitivity gps for precise indoor positioning. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 121–127. IEEE, 2008.

[90] Stephen Se, David G Lowe, and James J Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on robotics*, 21(3):364–375, 2005.

[91] Xuemin Shen and Li Deng. A dynamic system approach to speech enhancement using the h filtering algorithm. *IEEE Transactions on Speech and Audio Processing*, 7(4):391–399, 1999.

[92] Ali Asghar Nazari Shirehjini, Abdulsalam Yassine, and Shervin Shirmohammadi. An rfid-based position and orientation measurement system for mobile objects in

intelligent environments. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1664–1675, 2012.

[93] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004.

[94] Jakub Simanek, Michal Reinstein, and Vladimir Kubelka. Evaluation of the ekf-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Transactions on Mechatronics*, 20(2):985–990, 2015.

[95] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[96] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, Sep. 2004.

[97] A. Stancovici, S. Indreica, M.V. Micea, V. Cretu, and V. Groza. Relative localization methodology for autonomous robots in collaborative environments. In *Proc. IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1730–1733, Minneapolis, MN, USA, May 2013.

[98] D. Taniuchi, Xiaopeng Liu, D. Nakai, and T. Maekawa. Spring model based collaborative indoor position estimation with neighbor mobile devices. *IEEE Journal of Selected Topics in Signal Processing*, 9(2):268–277, Mar. 2015.

[99] Csaba D Toth, Joseph O'Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. CRC press, 2004.

[100] Rudolph Van Der Merwe. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD thesis, Oregon Health & Science University, 2004.

[101] Joel Vidal and Chyi-Yeu Lin. Simple and robust localization system using ceiling landmarks and infrared light. In *Control and Automation (ICCA), 2016 12th IEEE International Conference on*, pages 583–587. IEEE, 2016.

[102] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[103] J Wang et al. Pseudolite applications in positioning and navigation: Pro-gress and problems. *Positioning*, 1(03), 2002.

[104] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng. Efficient placement and dispatch of sensors in a wireless sensor network. *IEEE Trans. Mobile Comput.*, 7(2):262–274, Feb. 2008.

[105] Zidong Wang, Hongli Dong, Bo Shen, and Huijun Gao. Finite-horizon h_∞ filtering with missing measurements and quantization effects. *IEEE Transactions on Automatic Control*, 58(7):1707–1718, 2013.

[106] Wen-Tsai Huang, Chun-Lung Tsai, and Huei-Yung Lin. Mobile robot localization using ceiling landmarks and images captured from an RGB-D camera. In *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 855–860, Kachsiung, Taiwan, Jul. 2012.

[107] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.

[108] Fuwen Yang, Zidong Wang, Stanislao Lauria, and Xiaohui Liu. Mobile robot localization using robust extended H_∞ filtering. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 223(8):1067–1080, Dec. 2009.

[109] JaeMu Yun, EunTae Lyu, and JangMyung Lee. Image-based absolute positioning system for mobile robot navigation. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 261–269. Springer, 2006.

[110] Huijuan Zhang, Chengning Zhang, Wei Yang, and Chin-Yin Chen. Localization and navigation using qr code for mobile robot in indoor environment. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2501–2506. IEEE, 2015.

[111] J. Zhou, J. Shi, and X. Qu. Landmark placement for wireless localization in rectangular-shaped industrial facilities. *IEEE Trans. Veh. Technol.*, 59(6):3081–3090, Jul. 2010.

[112] Junyi Zhou and Jing Shi. Rfid localization algorithms and applicationsa review. *Journal of Intelligent Manufacturing*, 20(6):695–707, 2009.